Deep Limit Model-free Prediction in Regression

Kejin Wu

Department of Mathematics and Statistics
Loyola University Chicago

June 23, 2026

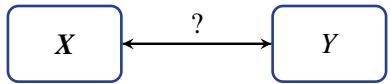
Joint work with

Dimitris Politis
University of California San Diego

Regression analysis

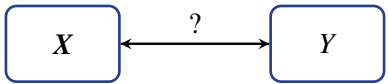
Regression analysis

Regression analysis is a statistical process to explore the relationship between the dependent/response variable Y and independent/predictors variable $\boldsymbol{X}$:



Regression analysis

Regression analysis is a statistical process to explore the relationship between the dependent/response variable Y and independent/predictors variable $\boldsymbol{X}$:

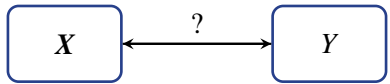


For example,

- Simple linear regression: relationship of heights between father and son;

Regression analysis

Regression analysis is a statistical process to explore the relationship between the dependent/response variable Y and independent/predictors variable $\boldsymbol{X}$:

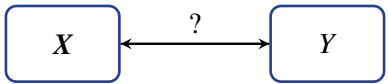


For example,

- Simple linear regression: relationship of heights between father and son;
- Quantile regression: impact of education, experience, etc., on different quantiles of income;

Regression analysis

Regression analysis is a statistical process to explore the relationship between the dependent/response variable Y and independent/predictors variable $\boldsymbol{X}$:



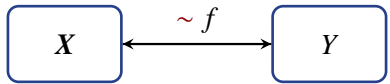
For example,

- Simple linear regression: relationship of heights between father and son;
- Quantile regression: impact of education, experience, etc., on different quantiles of income;
- Casual inference: effects of treatments on patients.

Model as bridge

Model as bridge

Classically, people assume there is a model f that may explain the relationship between $\boldsymbol{X}$ and Y:[1]



For example, a general homoscedastic model:

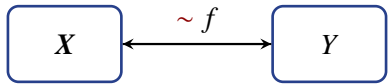
$$Y = f(\boldsymbol{X}) + \varepsilon;$$

where $f(\cdot)$ could be parametric or non-parametric; $\varepsilon \sim P_{\varepsilon}$.

[1] $\sim$ means that the association between $\boldsymbol{X}$ and Y may not be exactly described by f or there is a measurement error.

Model as bridge

Classically, people assume there is a model f that may explain the relationship between $\boldsymbol{X}$ and Y:[1]



For example, a general homoscedastic model:

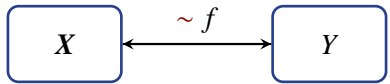
$$Y = f(\boldsymbol{X}) + \varepsilon;$$

where $f(\cdot)$ could be parametric or non-parametric; $\varepsilon \sim P_{\varepsilon}$.

[1] $\sim$ means that the association between $\boldsymbol{X}$ and Y may not be exactly described by f or there is a measurement error.

Model as bridge

Classically, people assume there is a model f that may explain the relationship between $\boldsymbol{X}$ and Y:[1]



For example, a general homoscedastic model:

$$Y = f(\boldsymbol{X}) + \varepsilon;$$

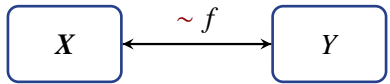
where $f(\cdot)$ could be parametric or non-parametric; $\varepsilon \sim P_{\varepsilon}$.

- Simple linear regression: $Y = \boldsymbol{\beta}^T \boldsymbol{X} + \varepsilon$;

[1] $\sim$ means that the association between $\boldsymbol{X}$ and Y may not be exactly described by f or there is a measurement error.

Model as bridge

Classically, people assume there is a model f that may explain the relationship between $\boldsymbol{X}$ and Y:[1]



For example, a general homoscedastic model:

$$Y = f(\boldsymbol{X}) + \varepsilon;$$

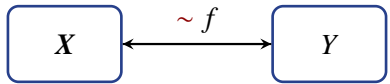
where $f(\cdot)$ could be parametric or non-parametric; $\varepsilon \sim P_\varepsilon$.

- Simple linear regression: $Y = \boldsymbol{\beta}^T \boldsymbol{X} + \varepsilon$;
- Quantile regression: $Q_Y(\tau|\boldsymbol{X}) = \boldsymbol{\beta}_\tau^T \boldsymbol{X}$;

[1] $\sim$ means that the association between $\boldsymbol{X}$ and Y may not be exactly described by f or there is a measurement error.

Model as bridge

Classically, people assume there is a model f that may explain the relationship between $\boldsymbol{X}$ and Y:[1]



For example, a general homoscedastic model:

$$Y = f(\boldsymbol{X}) + \varepsilon;$$

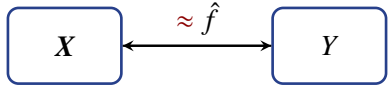
where $f(\cdot)$ could be parametric or non-parametric; $\varepsilon \sim P_{\varepsilon}$.

- Simple linear regression: $Y = \boldsymbol{\beta}^T \boldsymbol{X} + \varepsilon$;
- Quantile regression: $Q_Y(\tau|\boldsymbol{X}) = \boldsymbol{\beta}_\tau^T \boldsymbol{X}$;
- Casual inference: $f(\boldsymbol{x}) = \mathbb{E}\left(Y^1 - Y^0 \mid \boldsymbol{X} = \boldsymbol{x}\right)$ (Conditional Treatment Effects function).

[1] $\sim$ means that the association between $\boldsymbol{X}$ and Y may not be exactly described by f or there is a measurement error.

Estimation of model

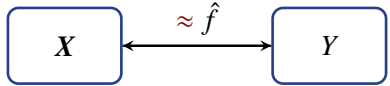
In practice, we estimate $f(\cdot)$ by $\hat{f}(\cdot)$ based on sample $\{\boldsymbol{X}_i, Y_i\}_{i=1}^n$: [2]



[2] Compared to $\sim$, $\approx$ involves additional estimation error.

Estimation of model

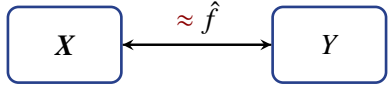
In practice, we estimate $f(\cdot)$ by $\hat{f}(\cdot)$ based on sample $\{\boldsymbol{X}_i, Y_i\}_{i=1}^n$: [2]



[2] Compared to $\sim$, $\approx$ involves additional estimation error.

Estimation of model

In practice, we estimate $f(\cdot)$ by $\hat{f}(\cdot)$ based on sample $\{\boldsymbol{X}_i, Y_i\}_{i=1}^n$: [2]



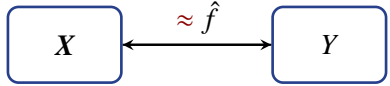
The estimation procedure can be done with some appropriate optimization criteria, e.g.,

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}\in\Theta^d} \frac{1}{n}\sum_{i=1}^{n}(Y_i - \boldsymbol{\beta}^T\boldsymbol{X}_i)^2 \text{ for simple linear regression; } \Theta^d \text{ is the parameter space.}$$

[2] Compared to $\sim$, $\approx$ involves additional estimation error.

Estimation of model

In practice, we estimate $f(\cdot)$ by $\hat{f}(\cdot)$ based on sample $\{\boldsymbol{X}_i, Y_i\}_{i=1}^n$: [2]



The estimation procedure can be done with some appropriate optimization criteria, e.g.,

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}\in\Theta^d} \frac{1}{n}\sum_{i=1}^{n}(Y_i - \boldsymbol{\beta}^T\boldsymbol{X}_i)^2 \text{ for simple linear regression; } \Theta^d \text{ is the parameter space.}$$

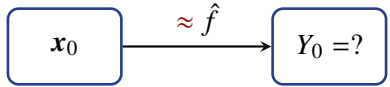
To quantify the estimation accuracy, we could build a Confidence Interval (CI).

[2] Compared to $\sim$, $\approx$ involves additional estimation error.

Prediction with model

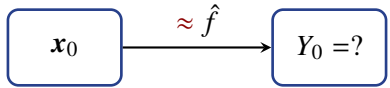
Prediction with model

We care about the prediction of Y_0 given some future value of $\boldsymbol{X}_0 = \boldsymbol{x}_0$ based on $\hat{f}(\cdot)$:



Prediction with model

We care about the prediction of Y_0 given some future value of $\boldsymbol{X}_0 = \boldsymbol{x}_0$ based on $\hat{f}(\cdot)$:



For simple linear regression, we take $\widehat{Y}_0 := \hat{\beta}^T \boldsymbol{x}_0$, which approximates L_2 optimal conditional prediction of Y, i.e.,

$$\widehat{Y}_0 \xrightarrow{p} \mathbb{E}(Y|\boldsymbol{x}_0) = \beta^T \boldsymbol{x}_0.$$

To quantify the prediction accuracy, we build Prediction Interval (PI) through:

To quantify the prediction accuracy, we build Prediction Interval (PI) through:

- (Normality assumption) Analytical way:

$$(Y_0 - \widehat{Y}_0)/\left(\hat{\sigma}\sqrt{1 + \boldsymbol{x}_0^T\left(\boldsymbol{X}_m^T\boldsymbol{X}_m\right)^{-1}\boldsymbol{x}_0}\right) \sim t_{n-d}\ ;\ \hat{\sigma} = \mathrm{RSS}/(n-d);\ \boldsymbol{X}_m \text{ is the design matrix.}$$

To quantify the prediction accuracy, we build Prediction Interval (PI) through:

- (Normality assumption) Analytical way:

$$(Y_0 - \widehat{Y}_0)/\left(\hat{\sigma}\sqrt{1 + \boldsymbol{x}_0^T\left(\boldsymbol{X}_m^T\boldsymbol{X}_m\right)^{-1}\boldsymbol{x}_0}\right) \sim t_{n-d}\ ;\ \hat{\sigma} = \mathrm{RSS}/(n-d);\ \boldsymbol{X}_m \text{ is the design matrix.}$$

- (Normality assumption fails) Simple plug-in method with empirical residual distribution:

$$[\widehat{Y}_0 + \widehat{F}_\epsilon^{-1}(\alpha/2), \widehat{Y}_0 + \widehat{F}_\epsilon^{-1}(1-\alpha/2)].$$

To quantify the prediction accuracy, we build Prediction Interval (PI) through:

- (Normality assumption) Analytical way:

$$(Y_0 - \widehat{Y}_0)/\left(\hat{\sigma}\sqrt{1 + \boldsymbol{x}_0^T\left(\boldsymbol{X}_m^T\boldsymbol{X}_m\right)^{-1}\boldsymbol{x}_0}\right) \sim t_{n-d}\ ;\ \hat{\sigma} = \mathrm{RSS}/(n-d);\ \boldsymbol{X}_m \text{ is the design matrix.}$$

- (Normality assumption fails) Simple plug-in method with empirical residual distribution:

$$[\widehat{Y}_0 + \widehat{F}_\epsilon^{-1}(\alpha/2), \widehat{Y}_0 + \widehat{F}_\epsilon^{-1}(1-\alpha/2)].$$

To quantify the prediction accuracy, we build Prediction Interval (PI) through:

- (Normality assumption) Analytical way:

$$(Y_0 - \widehat{Y}_0)/\left(\hat{\sigma}\sqrt{1 + \boldsymbol{x}_0^T\left(\boldsymbol{X}_m^T\boldsymbol{X}_m\right)^{-1}\boldsymbol{x}_0}\right) \sim t_{n-d}\ ;\ \hat{\sigma} = \mathrm{RSS}/(n-d);\ \boldsymbol{X}_m \text{ is the design matrix.}$$

- (Normality assumption fails) Simple plug-in method with empirical residual distribution:

$$[\widehat{Y}_0 + \widehat{F}_\epsilon^{-1}(\alpha/2), \widehat{Y}_0 + \widehat{F}_\epsilon^{-1}(1-\alpha/2)].$$

Limitation: The first one requires a normality assumption. The second one suffers from an undercoverage issue in the finite sample case.

Hard to determine the model

Essentially, all models are wrong, but some are useful.

—George Box

Hard to determine the model

> *Essentially, all models are wrong, but some are useful.*
>
> —*George Box*

Sometimes, parsimonious or simple models may work better than the true model for prediction purposes.

Basic assumptions for Model-free prediction

Goal: **Model-free** prediction without constraint model assumptions.

- $\boldsymbol{X}$ and Y have a joint distribution $P_{\boldsymbol{X},Y}$;[3]
- The domain of Y and X are compact sets, respectively, i.e., $\mathcal{Y} := [-M_1, M_1]$ and $\mathcal{X} := [-M_2, M_2]^d$; M_1 and M_2 are two positive constants. [4]

[3] We assume that the joint density of $P_{\boldsymbol{X},Y}$ exists to avoid some potential degenerate cases.

[4] A weaker assumption could be made such that $P(|Y| > \tau) \leq C\rho_1^{-\tau}$ and $P(\|\boldsymbol{X}\| > \tau) \leq C\rho_2^{-\tau}$ for some appropriate ρ_1 and ρ_2 (sub-exponential). Then, the event that $\boldsymbol{X}$ and Y belong to a compact set has a *high probability*.

Intuition

Intuition

In the standard regression context, we have the diagram, $\boldsymbol{X} \overset{\sim f}{\longleftrightarrow} Y$; $\sim$ is due to the model misspecification/insufficiency and unobserved measurement error.

Intuition

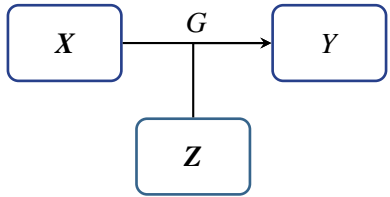
In the standard regression context, we have the diagram, $\boldsymbol{X} \overset{\sim f}{\longleftrightarrow} Y$; $\sim$ is due to the model misspecification/insufficiency and unobserved measurement error.

Can we **outsource** the unobserved error and make our model as flexible as it could?

Intuition

In the standard regression context, we have the diagram, $\boldsymbol{X} \overset{\sim f}{\longleftrightarrow} Y$; $\sim$ is due to the model misspecification/insufficiency and unobserved measurement error.

Can we **outsource** the unobserved error and make our model as flexible as it could?



Here, $G : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$; $\mathcal{Z}$ is the domain of the *reference* random variable $\boldsymbol{Z}$.

Noise outsourcing lemma

$G(\cdot, \cdot)$ could make a great connection between $\boldsymbol{X}$ and Y with the help of Z.

Noise outsourcing lemma

$G(\cdot, \cdot)$ could make a great connection between $\boldsymbol{X}$ and Y with the help of Z.

Lemma 1: Noise outsourcing (Bloem-Reddy et al., 2020)

Let $\boldsymbol{X}$ and Y be random variables with joint distribution $P_{\boldsymbol{X},Y}$. Then, there is a measurable function $G : [0, 1] \times \mathcal{X} \to \mathcal{Y}$ such that

$$(\boldsymbol{X}, Y) \overset{a.s.}{=} (\boldsymbol{X}, G(\boldsymbol{X}, Z)), \quad \text{where } Z \sim \text{Uniform}[0, 1] \text{ and } Z \perp\!\!\!\perp \boldsymbol{X}.$$

In particular, $Y \overset{a.s.}{=} G(\boldsymbol{X}, Z)$.

Noise outsourcing lemma

$G(\cdot, \cdot)$ could make a great connection between $\boldsymbol{X}$ and Y with the help of Z.

Lemma 1: Noise outsourcing (Bloem-Reddy et al., 2020)

Let $\boldsymbol{X}$ and Y be random variables with joint distribution $P_{\boldsymbol{X},Y}$. Then, there is a measurable function $G : [0, 1] \times \mathcal{X} \to \mathcal{Y}$ such that

$$(\boldsymbol{X}, Y) \overset{a.s.}{=} (\boldsymbol{X}, G(\boldsymbol{X}, Z)), \quad \text{where } Z \sim \text{Uniform}[0, 1] \text{ and } Z \perp\!\!\!\perp \boldsymbol{X}.$$

In particular, $Y \overset{a.s.}{=} G(\boldsymbol{X}, Z)$.

In other words, the randomness in the conditional distribution of Y given $\boldsymbol{X} = \boldsymbol{x}$ is outsourced to reference random variable Z through $G(\boldsymbol{x}, Z)$, where G is deterministic.

A continuous counterpart of $G(\cdot, \cdot)$

To estimate $G(\cdot, \cdot)$ with data, we hope it can possess some smoothness property (at least C^0).
Our proposition:

A continuous counterpart of $G(\cdot, \cdot)$

To estimate $G(\cdot, \cdot)$ with data, we hope it can possess some smoothness property (at least C^0). Our proposition:

Proposition 1: A continuous counterpart of $G(\cdot, \cdot)$ exists

Under our basic assumptions, there is a set D, and a continuous $\widetilde{G}(\cdot, \cdot) : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$ such that $\widetilde{G}(\boldsymbol{x}, z) = G(\boldsymbol{x}, z)$ for all $(\boldsymbol{x}, z) \in D \subseteq \mathcal{X} \times \mathcal{Z}$; here $\lambda((\mathcal{X} \times \mathcal{Z}) \backslash D) < \epsilon$ for $\forall \epsilon > 0$; λ denotes the Lebesgue measure; $\mathcal{Z}$ could be $\mathbb{R}^p$ or $[0, 1]^p$ if we take Z as $N(0, \boldsymbol{I}_p)$ or Uniform$[0, 1]^p$, respectively, for some positive integer p.

A continuous counterpart of $G(\cdot, \cdot)$

To estimate $G(\cdot, \cdot)$ with data, we hope it can possess some smoothness property (at least C^0). Our proposition:

Proposition 1: A continuous counterpart of $G(\cdot, \cdot)$ exists

Under our basic assumptions, there is a set D, and a continuous $\widetilde{G}(\cdot, \cdot) : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$ such that $\widetilde{G}(\boldsymbol{x}, z) = G(\boldsymbol{x}, z)$ for all $(\boldsymbol{x}, z) \in D \subseteq \mathcal{X} \times \mathcal{Z}$; here $\lambda((\mathcal{X} \times \mathcal{Z}) \backslash D) < \epsilon$ for $\forall \epsilon > 0$; λ denotes the Lebesgue measure; $\mathcal{Z}$ could be $\mathbb{R}^p$ or $[0, 1]^p$ if we take Z as $N(0, \boldsymbol{I}_p)$ or Uniform$[0, 1]^p$, respectively, for some positive integer p.

To simplify the notation, we will keep using $G(\cdot, \cdot)$ for this continuous counterpart. We will focus on estimating this continuous variant by DNN and then make predictions.

Estimation of $G(\cdot, \cdot)$

Estimation of $G(\cdot, \cdot)$

Define the population-level risk minimizer H_0 from a DNN class $\mathcal{F}_{\text{n,DNN}}$:

$$H_0 = \arg \min_{H \in \mathcal{F}_{n,\text{DNN}}} \mathbb{E}[(Y - H(\boldsymbol{X}, Z))]^2 = \arg \min_{H \in \mathcal{F}_{\text{n,DNN}}} \mathcal{R}(H);$$

Estimation of $G(\cdot, \cdot)$

Define the population-level risk minimizer H_0 from a DNN class $\mathcal{F}_{\text{n,DNN}}$:

$$H_0 = \arg \min_{H \in \mathcal{F}_{n,\text{DNN}}} \mathbb{E}[(Y - H(\boldsymbol{X}, Z))]^2 = \arg \min_{H \in \mathcal{F}_{\text{n,DNN}}} \mathcal{R}(H);$$

Recall that the risk for standard regression tasks is

$$\mathcal{R}(h) := \mathbb{E}[(Y - f(\boldsymbol{X}))^2].$$

Difference between traditional MSE risk

Table 1: Comparison between standard regression risk and our risk

	Geometry	σ-algebra
$\mathcal{R}(h)$	The risk minimizer is the projection of Y onto a closed subspace $\mathcal{S}_{\boldsymbol{X}}$ of L_2 consisting of all random variables which can be written in a function of $\boldsymbol{X}$.	$\mathbb{E}(Y\|\boldsymbol{X})$ is $\mathcal{D}_{\boldsymbol{X}}$-measurable.[5]
$\mathcal{R}(H)$	The risk minimizer is a projection of Y onto an extended version of $\mathcal{S}_{\boldsymbol{X}}$ by the reference variable Z.	$Y \overset{a.s.}{=} G(\boldsymbol{X}, Z)$ is $\mathcal{D}_{(\boldsymbol{X},Z)}$-measurable.

The noise outsourcing lemma implies an appropriate extension space such that we can find $G(X, Z) \overset{a.s.}{=} Y$ and then we anticipate $H_0(\cdot, \cdot)$ can be close to $G(\cdot, \cdot)$.

[5] $\mathcal{D}_{\boldsymbol{X}}$ is the σ-algebra generated by $\boldsymbol{X}$

Training algorithm

Training algorithm

Algorithm Training procedure to get empirically optimal estimator $\widehat{H}$

1: Initiate a DNN $H_{\boldsymbol{\theta}} \in \mathcal{F}_{\text{DNN}}$[6] and simulate $\{Z_i\}_{i=1}^n$ from P_Z.
2: **for** number of epochs **do**
3: Update $H_{\boldsymbol{\theta}}$ by descending its gradient with the chosen optimization algorithm:

$$\nabla_{\boldsymbol{\theta}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (Y_i - H_{\boldsymbol{\theta}}(\boldsymbol{X}_i, Z_i))^2 \right\}.$$

4: Clip the parameter of $H_{\boldsymbol{\theta}}$ to $[-m, m]$.
5: **end for**
6: **Return** The estimated $\widehat{H}(\cdot, \cdot)$.

[6] $\mathcal{F}_{\text{DNN}}$ is a user-chosen space that contains all DNN candidates.

Error bound for $\widehat{H}$

Error bound for $\widehat{H}$

Theorem 1: A high probability non-asymptotic error bound for $\widehat{H}$

Taking reference random variable $Z := \text{Uniform}[0,1]^p$ and $\mathcal{F}_{n,\text{DNN}}$ to be a class of fully connected feedforward DNN with width W_n and depth L_n.

When sample size n is large enough and under some further mild conditions, we have:

$$\left\|\widehat{H} - G\right\|^2_{L^2(X,Z)} \leq C \cdot n^{-\frac{2}{\tau+d+p}} + o(n^{-\frac{2}{\tau+d+p}}); \text{ for } \tau > 2; \tag{1}$$

with probability at least $1 - \exp(-n^{\frac{d+p}{\tau+d+p}})$; where C is a constant.

$W_n := 3^{d+p+3} \max\left\{(d+p)\left\lfloor N_1^{1/(d+p)}\right\rfloor, N_1+1\right\}$; $L_n := 12N_2 + 14 + 2(d+p)$; $N_1 = \left\lceil \frac{n^{\frac{d+p}{2(\tau+d+p)}}}{\log n} \right\rceil$; $N_2 = \lceil \log(n) \rceil$.

Estimation of conditional distribution

Estimation of conditional distribution

Define $\widehat{F}_{\widehat{H}(\boldsymbol{x}_0,Z)}$ as the empirical distribution of $\{\widehat{H}(\boldsymbol{x}_0, Z_i)\}_{i=1}^{S}$; S is the number of Monte Carlo sampling we apply to generate reference variable samples.

Estimation of conditional distribution

Define $\widehat{F}_{\widehat{H}(\boldsymbol{x}_0,Z)}$ as the empirical distribution of $\{\widehat{H}(\boldsymbol{x}_0, Z_i)\}_{i=1}^{S}$; S is the number of Monte Carlo sampling we apply to generate reference variable samples.

Under some additional restrictions about $P_{\boldsymbol{X},Y}$, we have

Theorem 2: Uniform estimation of $F_{Y|X}$ based on $\widehat{H}$

we have:

$$\sup_y \left| \widehat{F}_{\widehat{H}(\boldsymbol{x}_0,Z)}(y) - F_{Y|\boldsymbol{x}_0}(y) \right| \xrightarrow{p} 0, \text{ as } n \to \infty, S \to \infty,$$

for any $\boldsymbol{x}_0 \in \mathcal{X}$ expect a set with measure 0.

Other DNN generative methods

Recently, Zhou et al. (2023) and Liu et al. (2021) proposed two conditional generators to estimate the conditional distribution in the regression context. Their methods rely on the adversarial training strategy which was first proposed by Goodfellow et al. (2014). We use $\widehat{G}_{\mathrm{KL}}$ and $\widehat{G}_{\mathrm{WA}}$ to represent these two DNN-based deep generators, they can be trained by the below formula:

$$(\widehat{G}_{\mathrm{KL}}, \widehat{D}_{\mathrm{KL}}) = \arg \min_{G_\rho \in \mathcal{F}'_{\mathrm{DNN,G}}} \arg \max_{D_\phi \in \mathcal{F}'_{\mathrm{DNN,D}}} \frac{1}{n} \sum_{i=1}^{n} D_\phi(G_\rho(Z_i, X_i), X_i) - \frac{1}{n} \sum_{i=1}^{n} \exp(D_\phi(Y_i, X_i));$$

$$(\widehat{G}_{\mathrm{WA}}, \widehat{D}_{\mathrm{WA}}) = \arg \min_{G_\rho \in \mathcal{F}_{\mathrm{DNN,G}}} \arg \max_{D_\phi \in \mathcal{F}_{\mathrm{DNN,D}}} \frac{1}{n} \sum_{i=1}^{n} D_\phi(G_\rho(Z_i, X_i), X_i) - \frac{1}{n} \sum_{i=1}^{n} D_\phi(Y_i, X_i).$$

- The risk functions are based on variants of KL-divergence and Wasserstein-1 distance;
- D_ϕ is the discriminator/critic trained together with generator G_ρ adversarially.

Simulation setting for L_2 point prediction

We take the below model from Zhou et al. (2023) to generate n training and T test data:

$$Y_i = X_{i,1}^2 + \exp\left(X_{i,2} + X_{i,3}/3\right) + \sin\left(X_{i,4} + X_{i,5}\right) + \varepsilon_i;$$

where $\boldsymbol{X}_i$ and ε_i come from $N(0, \boldsymbol{I}_5)$ and $N(0, 1)$ truncated to $[-5, 5]^5$ and $[-5, 5]$, respectively.

Simulation setting for L_2 point prediction

We take the below model from Zhou et al. (2023) to generate n training and T test data:

$$Y_i = X_{i,1}^2 + \exp\left(X_{i,2} + X_{i,3}/3\right) + \sin\left(X_{i,4} + X_{i,5}\right) + \varepsilon_i;$$

where $\boldsymbol{X}_i$ and ε_i come from $N(0, \boldsymbol{I}_5)$ and $N(0, 1)$ truncated to $[-5, 5]^5$ and $[-5, 5]$, respectively.

We consider the L_2 point prediction: $\widehat{Y}_t = \frac{1}{S}\sum_{s=1}^{S} \widehat{\Pi}(\boldsymbol{x}_t, Z_s)$; $Z_s \sim N(0, I_p)$; $\boldsymbol{x}_t$ is the t-th observation of the test data; $\widehat{\Pi}$ represents trained model $\widehat{H}$, $\widehat{G}_{\mathrm{KL}}$ or $\widehat{G}_{\mathrm{WA}}$.

Simulation setting for L_2 point prediction

We take the below model from Zhou et al. (2023) to generate n training and T test data:

$$Y_i = X_{i,1}^2 + \exp(X_{i,2} + X_{i,3}/3) + \sin(X_{i,4} + X_{i,5}) + \varepsilon_i;$$

where $\boldsymbol{X}_i$ and ε_i come from $N(0, \boldsymbol{I}_5)$ and $N(0,1)$ truncated to $[-5,5]^5$ and $[-5,5]$, respectively.

We consider the L_2 point prediction: $\widehat{Y}_t = \frac{1}{S}\sum_{s=1}^{S} \widehat{\Pi}(\boldsymbol{x}_t, Z_s)$; $Z_s \sim N(0, I_p)$; $\boldsymbol{x}_t$ is the t-th observation of the test data; $\widehat{\Pi}$ represents trained model $\widehat{H}$, $\widehat{G}_{\mathrm{KL}}$ or $\widehat{G}_{\mathrm{WA}}$.

To measure different methods, we repeat the simulations K times and consider the error metric:

$$\widetilde{L} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{K}\sum_{k=1}^{K}(Y_{t,L_2} - \widehat{Y}_{k,t})^2;$$

where Y_{t,L_2} is the oracle L_2 optimal value of Y conditional on $\boldsymbol{x}_t$; $\widehat{Y}_{k,t}$ is the conditional L_2 point prediction based on the k-th training data.

We apply the same hyperparameter setting to train all related DNN.

We apply the same hyperparameter setting to train all related DNN.

We take $n = 2000$, $T = 2000$, $S = 10000$, $K = 200$ to compute the error metric.

We apply the same hyperparameter setting to train all related DNN.

We take $n = 2000$, $T = 2000$, $S = 10000$, $K = 200$ to compute the error metric.

For the benchmark method, we apply the numerical integration $\int_{\mathcal{Y}} y\hat{p}_{y|\boldsymbol{x}_t} dy$ with 1000 subdivisions to approximate $E(Y|\boldsymbol{x}_t)$; $\hat{p}_{y|\boldsymbol{x}_t}$ is the kernel conditional density estimator of Y conditional on $\boldsymbol{x}_t$.

Simulation results

Table 2: Point predictions of different methods.

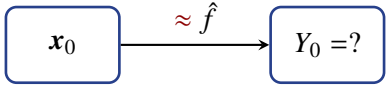
	$\widehat{H}$	$\widehat{G}_{\mathrm{KL}}$	$\widehat{G}_{\mathrm{WA}}$
SGD			
$p = 1$	**0.295**	3.793	4.709
$p = 3$	0.330	3.733	5.090
$p = 5$	0.358	3.753	5.405
$p = 10$	0.388	3.765	8.335
Adam			
$p = 1$	0.497	1.478	4.429
$p = 3$	0.319	1.652	2.772
$p = 5$	0.259	1.175	20.243
$p = 10$	**0.235**	1.259	1.883
RMSProp			
$p = 1$	0.249	0.707	0.900
$p = 3$	0.198	0.542	0.545
$p = 5$	**0.181**	0.344	0.402
$p = 10$	0.196	0.257	0.279

Note: The error metric $\widetilde{L}$ of using conditional kernel density estimation (i.e., the benchmark method CKDE) is around 0.767.

Motivation of pertinent prediction interval

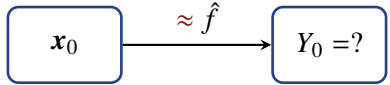
Motivation of pertinent prediction interval

The diagram to do prediction:



Motivation of pertinent prediction interval

The diagram to do prediction:

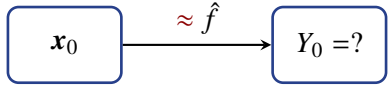


Here, $\approx$ represents error comes from two sources:

1. The association between $\boldsymbol{X}$ and Y is not exactly described by f or there is measurement error;
2. The estimation error within $\hat{f}$.

Motivation of pertinent prediction interval

The diagram to do prediction:



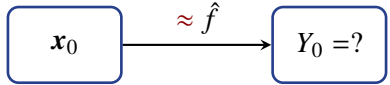
Here, $\approx$ represents error comes from two sources:

1. The association between $\boldsymbol{X}$ and Y is not exactly described by f or there is measurement error;
2. The estimation error within $\hat{f}$.

An oracle $G(\cdot, \cdot)$ can solve both error sources a.s.. However, error (2) still exists in practice when $\widehat{H}$ is applied.

Motivation of pertinent prediction interval

The diagram to do prediction:



Here, $\approx$ represents error comes from two sources:

1. The association between $\boldsymbol{X}$ and Y is not exactly described by f or there is measurement error;
2. The estimation error within $\hat{f}$.

An oracle $G(\cdot, \cdot)$ can solve both error sources a.s.. However, error (2) still exists in practice when $\widehat{H}$ is applied.

Try to make the Pertinent Prediction Interval to capture the error (2) in finite sample cases.

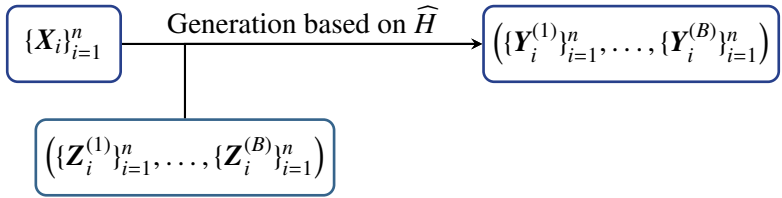
Basic idea of Pertinent PI (PPI)

Basic idea of Pertinent PI (PPI)

In the spirit of Bootstrap, we mimic the whole estimation process by pseudo values.

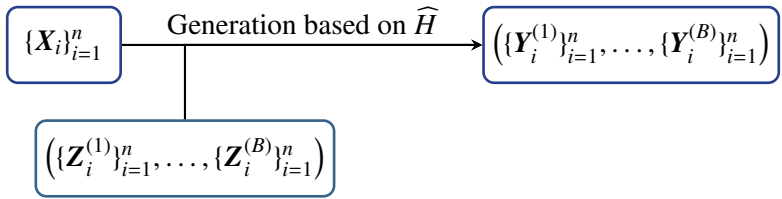
Basic idea of Pertinent PI (PPI)

In the spirit of Bootstrap, we mimic the whole estimation process by pseudo values.



Basic idea of Pertinent PI (PPI)

In the spirit of Bootstrap, we mimic the whole estimation process by pseudo values.



Train DNNs on pseudo values to get $\{\widehat{H}^{(b)}\}_{b=1}^B$ based on $\left(\{\boldsymbol{Y}_i^{(1)}\}_{i=1}^n, \ldots, \{\boldsymbol{Y}_i^{(B)}\}_{i=1}^n\right)$, $\{\boldsymbol{X}_i\}_{i=1}^n$ and $\{\boldsymbol{Z}_i\}_{i=1}^n$.

The form of PPI

Approximate the distribution of the predictive root R_0 by the variant R_0^* in the bootstrap world with $\{\widehat{H}^{(b)}\}_{b=1}^{B}$, i.e.,

$$R_0^* \xrightarrow[d]{Approximate} R_0;$$

where,

The form of PPI

Approximate the distribution of the predictive root R_0 by the variant R_0^* in the bootstrap world with $\{\widehat{H}^{(b)}\}_{b=1}^B$, i.e.,

$$R_0^* \xrightarrow[d]{Approximate} R_0;$$

where,

- R_0 could be $Y_0 - \widehat{Y}_{0,L_2}$; $Y_0 \sim P_{Y|\boldsymbol{x}_0}$ and $\widehat{Y}_{0,L_2} := \mathbb{E}(\widehat{H}(\boldsymbol{x}_0, Z))$ is the *estimated* optimal L_2 condition point prediction; we approximate it by $\frac{1}{S}\sum_{s=1}^S \widehat{H}(\boldsymbol{x}_0, Z_s)$;

The form of PPI

Approximate the distribution of the predictive root R_0 by the variant R_0^* in the bootstrap world with $\{\widehat{H}^{(b)}\}_{b=1}^B$, i.e.,

$$R_0^* \xrightarrow[d]{Approximate} R_0;$$

where,

- R_0 could be $Y_0 - \widehat{Y}_{0,L_2}$; $Y_0 \sim P_{Y|\boldsymbol{x}_0}$ and $\widehat{Y}_{0,L_2} := \mathbb{E}(\widehat{H}(\boldsymbol{x}_0, Z))$ is the *estimated* optimal L_2 condition point prediction; we approximate it by $\frac{1}{S}\sum_{s=1}^S \widehat{H}(\boldsymbol{x}_0, Z_s)$;
- R_0^* could be $Y_0^{(b)} - \widehat{Y}_{0,L_2}^{(b)}$; $Y_0^{(b)} \sim \widehat{H}(\boldsymbol{x}_0, Z)$ and $\widehat{Y}_{0,L_2}^{(b)} := \mathbb{E}(\widehat{H}^{(b)}(\boldsymbol{x}_0, Z))$ is the *estimated* optimal L_2 point prediction conditional on training data; we approximate it by $\frac{1}{S}\sum_{s=1}^S \widehat{H}^{(b)}(\boldsymbol{x}_0, Z_s)$; $\widehat{H}^{(b)}$ is the b-th re-estimation.

The form of PPI

Approximate the distribution of the predictive root R_0 by the variant R_0^* in the bootstrap world with $\{\widehat{H}^{(b)}\}_{b=1}^B$, i.e.,

$$R_0^* \xrightarrow[d]{Approximate} R_0;$$

where,

- R_0 could be $Y_0 - \widehat{Y}_{0,L_2}$; $Y_0 \sim P_{Y|\boldsymbol{x}_0}$ and $\widehat{Y}_{0,L_2} := \mathbb{E}(\widehat{H}(\boldsymbol{x}_0, Z))$ is the *estimated* optimal L_2 condition point prediction; we approximate it by $\frac{1}{S}\sum_{s=1}^S \widehat{H}(\boldsymbol{x}_0, Z_s)$;
- R_0^* could be $Y_0^{(b)} - \widehat{Y}_{0,L_2}^{(b)}$; $Y_0^{(b)} \sim \widehat{H}(\boldsymbol{x}_0, Z)$ and $\widehat{Y}_{0,L_2}^{(b)} := \mathbb{E}(\widehat{H}^{(b)}(\boldsymbol{x}_0, Z))$ is the *estimated* optimal L_2 point prediction conditional on training data; we approximate it by $\frac{1}{S}\sum_{s=1}^S \widehat{H}^{(b)}(\boldsymbol{x}_0, Z_s)$; $\widehat{H}^{(b)}$ is the b-th re-estimation.

The form of PPI

Approximate the distribution of the predictive root R_0 by the variant R_0^* in the bootstrap world with $\{\widehat{H}^{(b)}\}_{b=1}^B$, i.e.,

$$R_0^* \xrightarrow[d]{Approximate} R_0;$$

where,

- R_0 could be $Y_0 - \widehat{Y}_{0,L_2}$; $Y_0 \sim P_{Y|\boldsymbol{x}_0}$ and $\widehat{Y}_{0,L_2} := \mathbb{E}(\widehat{H}(\boldsymbol{x}_0, Z))$ is the *estimated* optimal L_2 condition point prediction; we approximate it by $\frac{1}{S}\sum_{s=1}^S \widehat{H}(\boldsymbol{x}_0, Z_s)$;
- R_0^* could be $Y_0^{(b)} - \widehat{Y}_{0,L_2}^{(b)}$; $Y_0^{(b)} \sim \widehat{H}(\boldsymbol{x}_0, Z)$ and $\widehat{Y}_{0,L_2}^{(b)} := \mathbb{E}(\widehat{H}^{(b)}(\boldsymbol{x}_0, Z))$ is the *estimated* optimal L_2 point prediction conditional on training data; we approximate it by $\frac{1}{S}\sum_{s=1}^S \widehat{H}^{(b)}(\boldsymbol{x}_0, Z_s)$; $\widehat{H}^{(b)}$ is the b-th re-estimation.

Thus, a pertinent PI with $1-\alpha$ coverage rate centered at $\widehat{Y}_{0,L_2}$ has the form:

$$\left[\widehat{Y}_{0,L_2} + Q_{\alpha/2}, \widehat{Y}_{0,L_2} + Q_{1-\alpha/2}\right];$$

$Q_{\alpha/2}$ and $Q_{1-\alpha/2}$ are $\alpha/2$ and $1-\alpha/2$ lower quantiles from the distribution of R_0^*, which can be approximated by the empirical distribution of $\{Y_0^{(b)} - \widehat{Y}_{0,L_2}^{(b)}\}_{b=1}^B$.

Unconditional coverage rate (CV) and average length (AL)

Table 3: Simulation results with varying n and p under nominal level 95% and RMSProp.

	CV	AL	*CV*	AL	*CV*	AL
p = 5	n = 200		n = 500		n = 2000	
QPI	0.861(0.170)	5.487(1.054)	0.927(0.110)	6.734(1.463)	0.787(0.177)	3.621(0.855)
PPI	0.893(0.139)	6.208(1.384)	0.941(0.095)	7.258(1.808)	0.789(0.173)	3.728(0.959)
PI-KL	0.842(0.193)	5.496(0.861)	0.869(0.157)	5.434(1.218)	0.913(0.104)	5.670(2.282)
PI-WA	0.852(0.181)	5.439(0.907)	0.882(0.150)	5.970(2.030)	0.899(0.105)	5.365(1.996)
p = 10						
QPI	0.928(0.129)	7.497(0.720)	0.949(0.094)	8.194(0.950)	0.855(0.157)	4.474(0.817)
PPI	**0.944(0.105)**	8.103(1.072)	0.961(0.076)	8.623(1.325)	0.855(0.154)	4.546(0.953)
PI-KL	0.900(0.133)	6.701(0.835)	0.925(0.119)	6.806(0.933)	0.928(0.099)	5.882(1.403)
PI-WA	0.898(0.146)	6.757(0.719)	0.933(0.116)	7.545(1.340)	0.934(0.100)	6.199(1.880)
p = 15						
QPI	0.915(0.137)	7.408(0.669)	0.945(0.097)	7.430(0.949)	0.915(0.123)	5.895(0.647)
PPI	0.930(0.119)	7.760(0.936)	**0.953(0.085)**	7.749(1.172)	0.916(0.121)	5.971(0.807)
PI-KL	0.909(0.136)	7.427(0.817)	0.949(0.095)	8.082(1.068)	0.943(0.089)	6.556(1.491)
PI-WA	0.901(0.137)	6.797(0.687)	0.950(0.095)	7.972(1.312)	0.947(0.088)	6.778(1.541)
p = 20						
QPI	0.879(0.172)	6.726(0.485)	0.959(0.085)	8.830(0.683)	0.940(0.102)	6.849(0.562)
PPI	0.893(0.154)	6.941(0.702)	0.966(0.073)	9.100(0.950)	0.942(0.097)	6.925(0.759)
PI-KL	0.923(0.126)	7.799(0.842)	0.954(0.087)	8.311(0.861)	0.946(0.093)	6.806(1.097)
PI-WA	0.910(0.140)	7.402(0.698)	0.945(0.099)	8.011(0.800)	0.946(0.092)	6.804(1.534)
p = 25						
QPI	0.871(0.172)	7.020(0.287)	0.961(0.088)	9.633(0.645)	0.946(0.099)	7.296(0.475)
PPI	0.884(0.160)	7.189(0.548)	0.967(0.078)	9.881(0.938)	**0.948(0.095)**	7.370(0.695)
PI-KL	0.907(0.142)	7.370(0.618)	0.954(0.090)	8.670(0.813)	0.945(0.093)	6.915(1.009)
PI-WA	0.897(0.151)	7.071(0.510)	0.960(0.081)	8.514(0.942)	0.944(0.097)	7.117(1.491)

Simulation results of (conditional) CV: PPI vs PI-KL

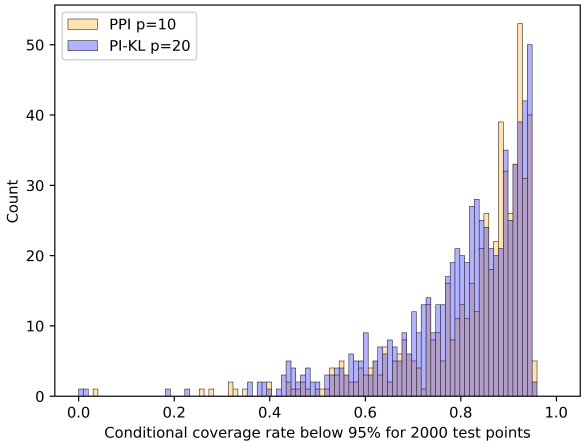


Figure 1: Histograms of all undercoverage CV_2 (CV_2 less than nominal level 95%) of PPI and PI-KL when $n = 200$.

Summary

- Instead of assuming a regression model, we propose a deep generative method to do regression based on the noise outsourcing lemma. The training procedure is straightforward.
- We design the prediction algorithm to capture the model estimation variability so that the prediction coverage can be improved with finite samples.
- The performance of our method is better than other two deep generative methods and also the conditional conformal prediction in Gibbs et al. (2025).
- The theoretical understanding of pertinence based on the convolution involved in the predictive root.

Summary

- Instead of assuming a regression model, we propose a deep generative method to do regression based on the noise outsourcing lemma. The training procedure is straightforward.
- We design the prediction algorithm to capture the model estimation variability so that the prediction coverage can be improved with finite samples.
- The performance of our method is better than other two deep generative methods and also the conditional conformal prediction in Gibbs et al. (2025).
- The theoretical understanding of pertinence based on the convolution involved in the predictive root.

Limitations:

- The dimension of the reference variable is a tuning parameter, which needs to be determined in practice.
- Although the training of DNN on pseudo data can be parallelized, the computation is still heavy in terms of memory.

Thank you!

References

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

Bloem-Reddy, B., Whye, Y., et al. (2020). Probabilistic symmetries and invariant neural networks. *Journal of Machine Learning Research*, 21(90):1–61.

Gibbs, I., Cherian, J. J., and Candès, E. J. (2025). Conformal prediction with conditional guarantees. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 87(4):1100–1126.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Liu, S., Zhou, X., Jiao, Y., and Huang, J. (2021). Wasserstein generative learning of conditional distribution. *arXiv preprint arXiv:2112.10039*.

Pang, T., Yang, X., Dong, Y., Su, H., and Zhu, J. (2020). Bag of tricks for adversarial training. *arXiv preprint arXiv:2010.00467*.

Zhou, X., Jiao, Y., Liu, J., and Huang, J. (2023). A deep generative approach to conditional sampling. *Journal of the American Statistical Association*, 118(543):1837–1848.

Intuition behind our Deep limit model-free prediction algorithm

We provide a toy example to explain the motivation of our training procedure.

Remark: An illustration example

Suppose we need to estimate the coefficient β of a linear regression model $Y = \beta^T \cdot \boldsymbol{X} + \epsilon$ with a fixed design based on samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$; here, ϵ has zero mean and finite variance.

- OLS: $\widehat{\beta} := \arg\min_\beta \frac{1}{n} \sum_{i=1}^n (y_i - \beta_i^T \cdot \boldsymbol{x}_i)^2$ which is consistent under standard conditions.
- Variant of OLS: $\widehat{\beta}^* := \arg\min_\beta \frac{1}{n} \sum_{i=1}^n (y_i - (\beta_i^T \cdot \boldsymbol{x}_i + \epsilon_i^*))^2$ where $\{\epsilon_i^*\}_{i=1}^n$ are independent of $\boldsymbol{X}$ and can be generated from any distribution with mean zero and finite variance.

$\widehat{\beta}^*$ is also consistent although $\widehat{\beta}$ would generally be more efficient.

Analogously, our DNN-based estimation $\widehat{H}^*$ converges to H_0 in the mean square sense even using the artificially generated $\{Z_i^*\}_{i=1}^n$.

Preliminary comparisons

Table 4: Comparison between different DNN-based methods

	$\widehat{H}$	$\widehat{G}_{\mathrm{KL}}, \widehat{G}_{\mathrm{WA}}$
Stability	The training process is more stable and directly due to the MSE-like loss function.	The training process is sensitive to the training setting and depends on D_ϕ being optimal given current step G_ρ.
Metrics	The optimization corresponds to minimizing the Kolmogorov distance between two distributions.	The optimization corresponds to minimizing KL-divergence and Wasserstein-1 distance[7].
Computability	Only one DNN need to be trained.	Two DNNs need to be trained adversarially.

[7]The "distance" between two distributions converges to 0 under the metric of Wasserstein-1 distance or KL-divergence implies the convergence measured by Kolmogorov distance.

Hyperparameter setting

We apply the same hyperparameter setting to train $\widehat{H}$, $\widehat{G}_{\mathrm{KL}}$ and $\widehat{G}_{\mathrm{WA}}$: $n = 2000$; $T = 2000$; $S = 10000$; $K = 200$; $p = 1, 3, 5, 10$, $m = 20$; Learning rate: 0.001; Number of epochs: 10000.

For the optimizer of the adversarial training process, Arjovsky et al. (2017) proposed using optimizer RMSProp with Wasserstein distance is more appropriate. However, Pang et al. (2020) argued that SGD-based optimizers are better. We consider three common optimizers, SGD, Adam and RMSProp.

KL-divergence and Wasserstein-1 distance

- KL-divergence: if f, g are densities of the measures μ, ν with respect to a dominating measure λ,

$$d_I(\mu, \nu) := \int_{S(\mu)} f \log(f/g) d\lambda.$$

  where $S(\mu)$ is the support of μ on Ω.

- Wasserstein-1 distance: for $\Omega = \mathbb{R}$, if F, G are the distribution functions of μ, ν respectively, the Kantorovich metric is defined by

$$\begin{aligned} d_W(\mu, \nu) &:= \int_{-\infty}^{\infty} |F(x) - G(x)| dx \\ &= \int_0^1 \left| F^{-1}(t) - G^{-1}(t) \right| dt. \end{aligned}$$

Theoretical explanations of PPI

Under further assumptions about the joint distribution $P_{X,Y}$, we have:

Theorem 3: Theoretical understanding of PPI with DNN

For an appropriate sequence of sets Ω_n, such that $\mathbb{P}((\{\boldsymbol{X}_i, Y_i, Z_i\}_{i=1}^n) \notin \Omega_n) = o(1)$, PPI can capture the estimation variability under $S \to \infty$ in an appropriate rate for each n, when $n \to \infty$. Furthermore,

$$\sup_y \left| \widehat{F}_{\widehat{H}(\boldsymbol{x}_0,Z)} \star \phi_\sigma(y) - F_{Y|\boldsymbol{x}_0} \star \phi_\sigma(y) \right| \leq \sup_y \left| \widehat{F}_{\widehat{H}(\boldsymbol{x}_0,Z)}(y) - F_{Y|\boldsymbol{x}_0}(y) \right| \text{ with probability 1;}$$

$\widehat{F}_{\widehat{H}(\boldsymbol{x}_0,Z)}$ is the empirical distribution of $\{\widehat{H}(\boldsymbol{x}_0, Z_i)\}_{i=1}^S$; $\star$ is the convolution operator; ϕ_σ is the density function of the normal distribution $N(0, \sigma^2)$.

Remark of Theorem 6

- **PPI can capture the estimation variability**: Since the distribution of R_0^* can approximate the distribution of R_0, PPI captures the estimation variability in finite sample cases to some extent.
- **A convolution implied in predictive root**: It comes from rewriting the predictive root as $R_0 := Y_0 - \mathbb{E}(Y_0|\boldsymbol{x}_0) + \mathbb{E}(Y_0|\boldsymbol{x}_0) - \widehat{Y}_{0,L_2}$; $Y_0 - \mathbb{E}(Y_0|\boldsymbol{x}_0)$ only depends on $P_{Y|\boldsymbol{x}_0}$ and $\mathbb{E}(Y_0|\boldsymbol{x}_0) - \widehat{Y}_{0,L_2}$ is a (asymptotically shrinking) Gaussian distribution. Thus the below inequality from the previous theorem reveals that we need less data to achieve the same accuracy of the distribution estimation under this convolution approach.

$$\sup_y \left| \widehat{F}_{\widehat{H}(\boldsymbol{x}_0,Z)} \star \phi_\sigma(y) - F_{Y|\boldsymbol{x}_0} \star \phi_\sigma(y) \right| \leq \sup_y \left| \widehat{F}_{\widehat{H}(\boldsymbol{x}_0,Z)}(y) - F_{Y|\boldsymbol{x}_0}(y) \right| \text{ with probability 1.}$$