Deep Limit Model-free Prediction in Regression

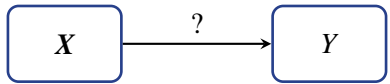
Kejin Wu [1]
(joint work with Dimitris Politis [2])

1. Department of Mathematics and Statistics, Loyola University Chicago
2. Department of Mathematics, University of California San Diego

December 16, 2025
ICSDS 2025

Regression analysis

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

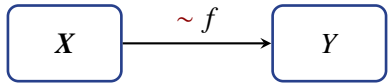


For example,

- 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

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



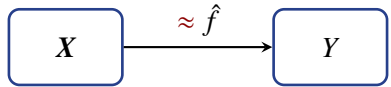
For example,

- 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]

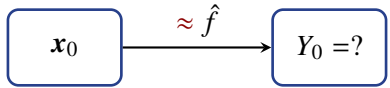


To quantify the estimation accuracy, we could build a Confidence Interval (CI).

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

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:

- (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: Require the normality assumption, otherwise undercoverage in the finite sample case.

What if model is wrong?

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

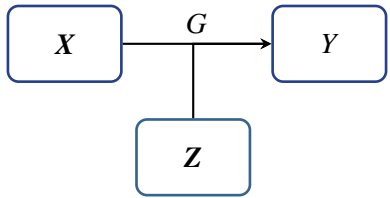
—George Box

Thus, we propose our method without restrictive model assumptions.

Intuition

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

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 *perfect* connection between $\boldsymbol{X}$ and Y.

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 $G(\cdot, \cdot)$

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.

Deep Neural Networks (DNN) estimator

The estimation error of a DNN estimator $\widehat{H}$ can be decomposed into two sources:

(1) The stochastic error, which measures the difference between $\widehat{H}$ and the best estimator H^* in a DNN class $\mathcal{F}_{\text{DNN}}$; $H^* := \underset{H \in \mathcal{F}_{\text{DNN}}}{\arg\min} \left\| \widetilde{G} - H \right\|_\infty$;

(2) The approximation error, which measures the difference between $\widetilde{G}$ and H^* in a DNN class $\mathcal{F}_{\text{DNN}}$.

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 samples.

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

Theorem 1: 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 $\boldsymbol{x}_0 \in \mathcal{X}$.

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 objective 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;
- $\mathcal{F}_{\cdot,\cdot}$ and $\mathcal{F}'_{\cdot,\cdot}$ represent appropriate DNN classes.

Simulation setting for optimal L_2 point prediction

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

$$Y_i = X_{i,1}^2 + \exp\left(X_{i,2} + X_{i,3}/3\right) + X_{i,4} - X_{i,5} + \left(0.5 + X_{i,2}^2/2 + X_{i,5}^2/2\right) \cdot \varepsilon_i;$$

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 apply the same hyperparameter setting to train all DNN.

For the structure of DNN, we separate the simulation studies into two groups.

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

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

Simulation results

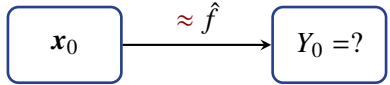
Table 1: Point predictions of different methods under groups (a) and (b).

	Group (a)			Group (b)		
	$\widehat{H}$	$\widehat{G}_{\mathrm{KL}}$	$\widehat{G}_{\mathrm{WA}}$	$\widehat{H}$	$\widehat{G}_{\mathrm{KL}}$	$\widehat{G}_{\mathrm{WA}}$
SGD						
$p = 1$	0.309	3.931	10.39	0.292	3.827	82.97
$p = 3$	0.298	4.009	11.10	0.285	3.762	56644
$p = 5$	0.296	4.036	40.39	0.281	3.801	12843
$p = 10$	**0.294**	4.116	182.3	**0.280**	3.812	11378
Adam						
$p = 1$	1.608	1.838	3558	1.572	1.836	14322
$p = 3$	0.832	1.105	8.480	0.843	1.549	43.48
$p = 5$	0.604	0.820	43.85	0.591	1.166	43.84
$p = 10$	**0.412**	0.495	5.523	**0.422**	0.817	14.50
RMSProp						
$p = 1$	0.960	1.767	1.910	0.973	1.620	2.326
$p = 3$	0.601	1.049	1.248	0.597	0.964	1.263
$p = 5$	0.484	0.779	0.908	0.479	0.727	0.903
$p = 10$	**0.365**	0.463	0.598	**0.352**	0.494	0.508

Note: The prediction error of using conditional kernel density estimation is around 1.210.

Motivation to make Pertinent Prediction Interval

Recall the diagram:



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

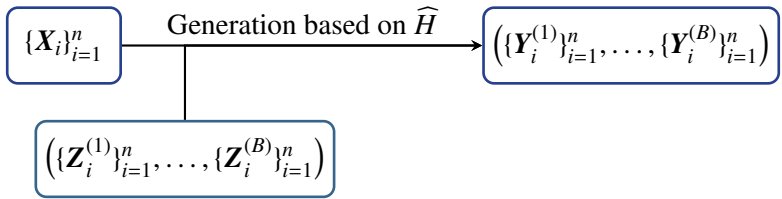
1. The association between $\boldsymbol{X}$ and Y is not exactly described by f or there is measurement error;
2. The estimation error within $\hat{f}$.

An oracle $G(\cdot, \cdot)$ can solve both error sources a.s. However, error (2) still exists in practice.

Thus, we attempt to build the Pertinent Prediction Interval (PPI), which can capture the estimation variability in finite sample cases.

Preparations for PPI

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



Them, make re-estimation 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 based on $\widehat{H}$

The fundamental idea of building PPI: approximate the predictive root R_0 by the variant R_0^* in the bootstrap world, i.e., conditional on training data $\{(\boldsymbol{X}_i, Y_i, Z_i)\}_{i=1}^n$:

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

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 of $P_{R_0^*}$, the distribution of R_0^*. In practice, $P_{R_0^*}$ can be approximated by the empirical distribution of $\{Y_0^{(b)} - \widehat{Y}_{0,L_2}^{(b)}\}_{b=1}^B$.

Simulation results of coverage rate of different PIs

Table 2: Simulation results of CV_1 with varying n and p.

	CV_1	AL	CV_1	AL	CV_1	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)

Thank you!

See more details on theory and real-data analyses from the paper:

Deep Limit Model-free Prediction in Regression

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.

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.

Backup: Additional slides

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 $\{Z_i^*\}_{i=1}^n$.

Difference between traditional MSE risk

Recall that the risk for standard regression tasks is

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

Table 3: Comparison between standard regression risk and our risk

	Geometry	σ-algebra
$\mathcal{R}_s$	The optimal estimator 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.[3]
$\mathcal{R}$	The optimal estimator is a projection of Y onto an extended version of $\mathcal{S}_{\boldsymbol{X}}$ by random variable Z.	$Y \overset{a.s.}{=} G(\boldsymbol{X}, Z)$ is $\mathcal{D}_{(\boldsymbol{X},Z)}$-measurable.

[3] $\mathcal{D}_{\boldsymbol{X}}$ is the σ-algebra generated by $\boldsymbol{X}$; $\mathbb{E}(Y|\boldsymbol{X})$ could also equal to Y a.s. if Y is $\mathcal{D}_{\boldsymbol{X}}$-measurable, e.g., $\mathbb{E}(Y|Y) = Y$.

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[4].
Computability	Only one DNN need to be trained.	Two DNNs need to be trained adversarially.

[4]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

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

Simulation results of conditional coverage rate: PPI vs PI-KL

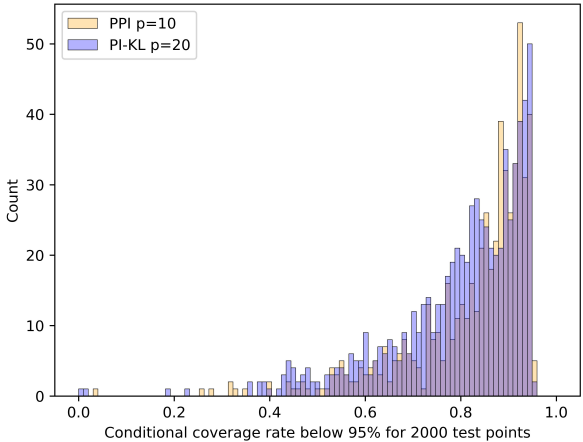


Figure 1: Histograms of all undercoverage CV_2 (CV_2 less than nominal level 95%) of PPI and PI-KL.

Simulation results of conditional coverage rate: PPI vs PI-WA

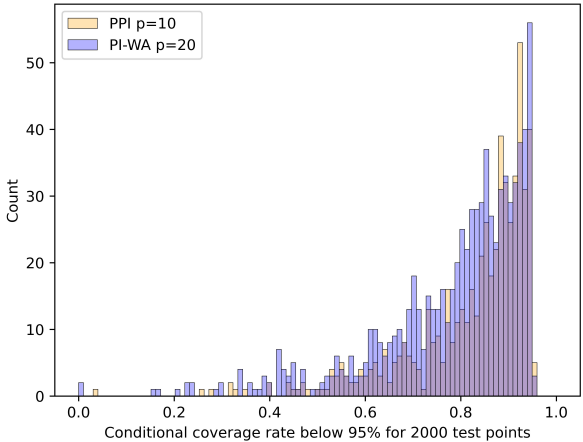


Figure 2: Histograms of all undercoverage CV_2 (CV_2 less than nominal level 95%) of PPI and PI-WA.

Simulation results of conditional coverage rate: PPI vs QPI

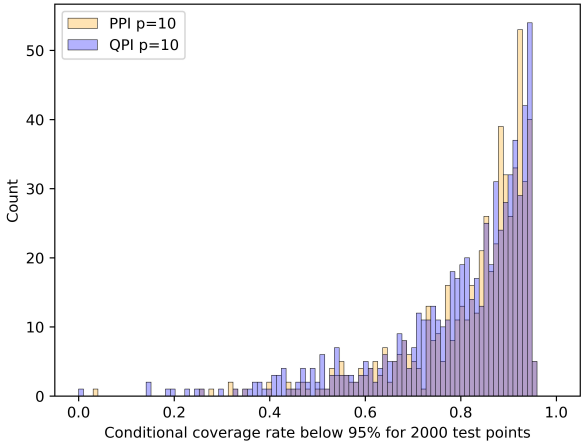


Figure 3: Histograms of all undercoverage CV_2 (CV_2 less than nominal level 95%) of PPI and QPI.

Error bound for $\widehat{H}$

Theorem 2: A high probability non-asymptotic error bound for $\widehat{H}$

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

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

$$\left\|\widehat{H} - H_0\right\|^2_{L^2(X,Z)} \leq C \cdot n^{-\frac{2}{\tau+d+p}} + o(n^{-\frac{2}{\tau+d+p}}); \text{ for } d+p \geq 2; \tau > 2; \tag{1}$$

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

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

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 3

- **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.

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