

CARD: CLASSIFICATION AND REGRESSION DIFFUSION MODELS

XIZEWEN HAN, HUANGJIE ZHENG & MINGYUAN ZHOU

Department of Statistics and Data Sciences & McCombs School of Business, The University of Texas at Austin

Modeling $P(y | x)$ instead of $\mathbb{E}(y | x)$

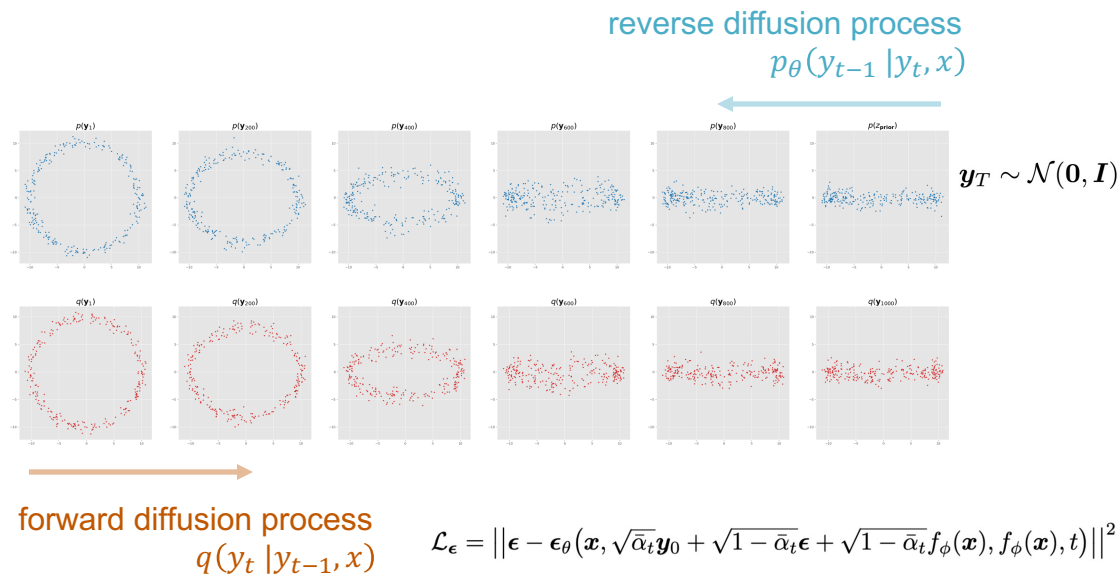
How do we model $P(y | x)$, the *conditional distribution* of a continuous or categorical response variable y given its covariates x , if we are interested in more than a point estimate of the conditional mean $\mathbb{E}(y | x)$?

Potential scenarios:

- Uncertainty estimation plays an important role for the problem in hand
- $P(y | x)$ is multi-modal, *e.g.*, when there are missing covariates in x

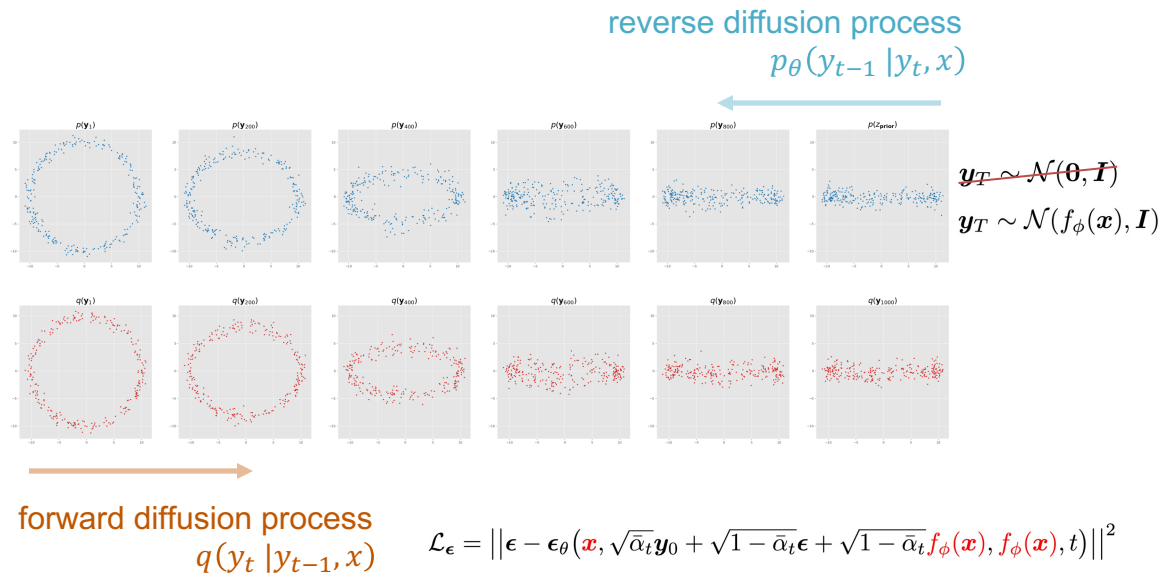
Modeling $P(y | x)$ instead of $\mathbb{E}(y | x)$

- Denoising Diffusion Probabilistic Models (Ho et al., 2020)

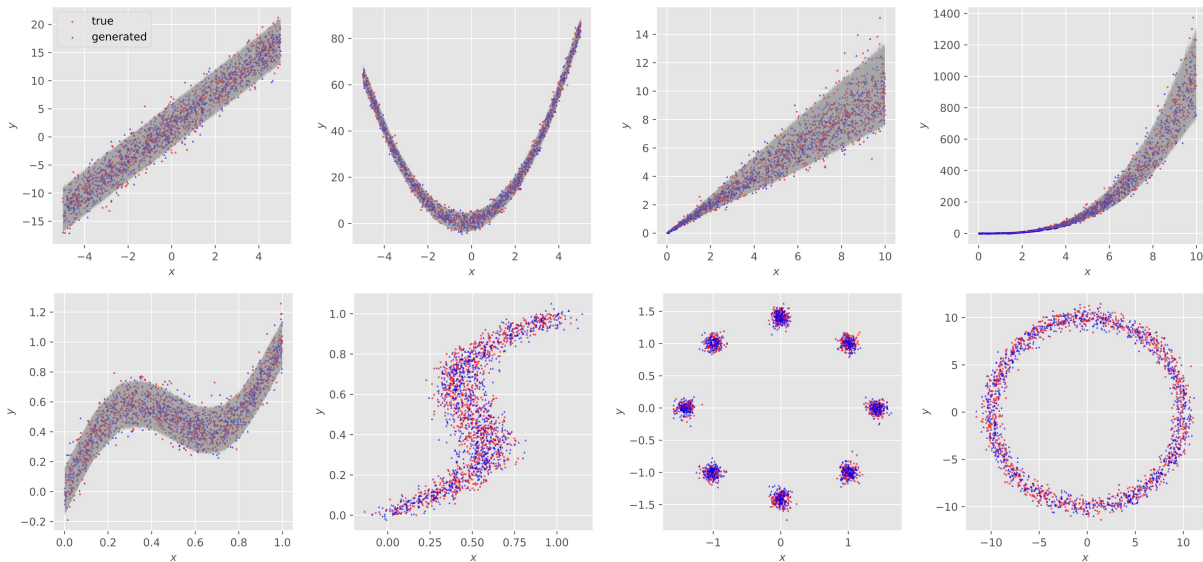


Modeling $P(y | x)$ instead of $\mathbb{E}(y | x)$

- CARD



Regression



Regression toy example scatter plots.

(**Top**) left to right: linear regression, quadratic regression, log-log linear regression, log-log cubic regression;
 (**Bottom**) left to right: sinusoidal regression, inverse sinusoidal regression, 8 Gaussians, full circle.

Regression

Dataset	RMSE ↓				
	PBP	MC Dropout	Deep Ensembles	GCDS	CARD (ours)
Boston	2.89 ± 0.74	3.06 ± 0.96	3.17 ± 1.05	2.75 ± 0.58	2.61 ± 0.63
Concrete	5.55 ± 0.46	5.09 ± 0.60	4.91 ± 0.47	5.39 ± 0.55	4.77 ± 0.46
Energy	1.58 ± 0.21	1.70 ± 0.22	2.02 ± 0.32	0.64 ± 0.09	0.52 ± 0.07
Kin8nm ¹	9.42 ± 0.29	7.10 ± 0.26	8.65 ± 0.47	8.88 ± 0.42	6.32 ± 0.18
Naval ²	0.41 ± 0.08	0.08 ± 0.03	0.09 ± 0.01	0.14 ± 0.05	0.02 ± 0.00
Power	4.10 ± 0.15	4.04 ± 0.14	4.02 ± 0.15	4.11 ± 0.16	3.93 ± 0.17
Protein	4.65 ± 0.02	4.16 ± 0.12	4.45 ± 0.02	4.50 ± 0.02	3.73 ± 0.01
Wine	0.64 ± 0.04	0.62 ± 0.04	0.63 ± 0.04	0.66 ± 0.04	0.63 ± 0.04
Yacht	0.88 ± 0.22	0.84 ± 0.27	1.19 ± 0.49	0.79 ± 0.26	0.65 ± 0.25
Year	8.86 ± NA	8.77 ± NA	8.79 ± NA	9.20 ± NA	8.70 ± NA
# best	0	1	0	0	9

Dataset	NLL ↓				
	PBP	MC Dropout	Deep Ensembles	GCDS	CARD (ours)
Boston	2.53 ± 0.27	2.46 ± 0.12	2.35 ± 0.16	18.66 ± 8.92	2.35 ± 0.12
Concrete	3.19 ± 0.05	3.21 ± 0.18	2.93 ± 0.12	13.64 ± 6.88	2.96 ± 0.09
Energy	2.05 ± 0.05	1.50 ± 0.11	1.40 ± 0.27	1.46 ± 0.72	1.04 ± 0.06
Kin8nm	-0.83 ± 0.02	-1.14 ± 0.05	-1.06 ± 0.02	-0.38 ± 0.36	-1.32 ± 0.02
Naval	-3.97 ± 0.10	-4.45 ± 0.38	-5.94 ± 0.10	-5.06 ± 0.48	-7.54 ± 0.05
Power	2.92 ± 0.02	2.90 ± 0.03	2.89 ± 0.02	2.83 ± 0.06	2.82 ± 0.02
Protein	3.05 ± 0.00	2.80 ± 0.08	2.89 ± 0.02	2.81 ± 0.09	2.49 ± 0.03
Wine	1.03 ± 0.03	0.93 ± 0.06	0.96 ± 0.06	6.52 ± 21.86	0.92 ± 0.05
Yacht	1.58 ± 0.08	1.73 ± 0.22	1.11 ± 0.18	0.61 ± 0.34	0.90 ± 0.08
Year	3.69 ± NA	3.42 ± NA	3.44 ± NA	3.43 ± NA	3.34 ± NA
# best	0	0	1	1	8

Dataset	QICE ↓				
	PBP	MC Dropout	Deep Ensembles	GCDS	CARD (ours)
Boston	3.50 ± 0.88	3.82 ± 0.82	3.37 ± 0.00	11.73 ± 1.05	3.45 ± 0.83
Concrete	2.52 ± 0.60	4.17 ± 1.06	2.68 ± 0.64	10.49 ± 1.01	2.30 ± 0.66
Energy	6.54 ± 0.90	5.22 ± 1.02	3.62 ± 0.58	7.41 ± 2.19	4.91 ± 0.94
Kin8nm	1.31 ± 0.25	1.50 ± 0.32	1.17 ± 0.22	7.73 ± 0.80	0.92 ± 0.25
Naval	4.06 ± 1.25	12.50 ± 1.95	6.64 ± 0.60	5.76 ± 2.25	0.80 ± 0.21
Power	0.82 ± 0.19	1.32 ± 0.37	1.09 ± 0.26	1.77 ± 0.33	0.92 ± 0.21
Protein	1.69 ± 0.09	2.82 ± 0.41	2.17 ± 0.16	2.33 ± 0.18	0.71 ± 0.11
Wine	2.22 ± 0.64	2.79 ± 0.56	2.37 ± 0.63	3.13 ± 0.79	3.39 ± 0.69
Yacht	6.93 ± 1.74	10.33 ± 1.34	7.22 ± 1.41	5.01 ± 1.02	8.03 ± 1.17
Year	2.96 ± NA	2.43 ± NA	2.56 ± NA	1.61 ± NA	0.53 ± NA
# best	2	0	2	1	5

QICE: the mean absolute error between the proportion of true data contained by each quantile interval of the generated y samples and the optimal proportion $1/M$.

$$\text{QICE} := \frac{1}{M} \sum_{m=1}^M \left| r_m - \frac{1}{M} \right|, \text{ where } r_m = \frac{1}{N} \sum_{n=1}^N \mathbb{1}_{y_n \geq \hat{y}_n^{\text{low}_m}} \cdot \mathbb{1}_{y_n \leq \hat{y}_n^{\text{high}_m}}.$$

Evaluation metric tables of UCI regression tasks.
 (Top to Bottom) RMSE, NLL, and QICE.

Classification

Algorithm 1 Training (Regression)

- 1: Pre-train $f_\phi(\mathbf{x})$ that predicts $\mathbb{E}(\mathbf{y} \mid \mathbf{x})$ with MSE
- 2: **repeat**
- 3: Draw $y_0 \sim q(\mathbf{y}_0 \mid \mathbf{x})$
- 4: Draw $t \sim \text{Uniform}(\{1 \dots T\})$
- 5: Draw $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 6: Compute noise estimation loss

$$\mathcal{L}_\epsilon = \left\| \epsilon - \epsilon_\theta(\mathbf{x}, \sqrt{\bar{\alpha}_t} \mathbf{y}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon + \sqrt{1 - \bar{\alpha}_t} f_\phi(\mathbf{x}), f_\phi(\mathbf{x}), t) \right\|^2$$

- 7: Take numerical optimization step on:

$$\nabla_\theta \mathcal{L}_\epsilon$$

- 8: **until** Convergence
-

Algorithm 2 Inference (Regression)

- 1: $\mathbf{y}_T \sim \mathcal{N}(f_\phi(\mathbf{x}), \mathbf{I})$
 - 2: **for** $t = T$ to 1 **do**
 - 3: Draw $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$
 - 4: Calculate reparameterized $\hat{\mathbf{y}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{y}_t - (1 - \sqrt{\bar{\alpha}_t}) f_\phi(\mathbf{x}) - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(\mathbf{x}, \mathbf{y}_t, f_\phi(\mathbf{x}), t) \right)$
 - 5: Let $\mathbf{y}_{t-1} = \gamma_0 \hat{\mathbf{y}}_0 + \gamma_1 \mathbf{y}_t + \gamma_2 f_\phi(\mathbf{x}) + \sqrt{\tilde{\beta}_t} \mathbf{z}$ if $t > 1$, else set $\mathbf{y}_{t-1} = \hat{\mathbf{y}}_0$
 - 6: **end for**
 - 7: **return** \mathbf{y}_0
-

Adaptation for classification tasks:

1. For \mathbf{y}_0 , replace the response variable with a one-hot encoded label vector;
2. For $f_\phi(\mathbf{x})$, replace the mean estimator with a classifier pre-trained with the cross-entropy objective, which outputs softmax probabilities of the class labels.

Classification

- Assess model prediction confidence *at the instance level*

For each test instance, we sample N class prototype reconstructions by CARD, and perform the following computations:

1. We directly calculate the prediction interval width (PIW) between the 2.5th and 97.5th percentiles of the N reconstructed values for all classes, *i.e.*, with C different classes in total, we would obtain C PIWs for each instance;
2. We then convert the samples into probability space as a softmax form of a temperature-weighted Brier score, and apply paired two-sample t -test as an uncertainty estimation method: we obtain the 1st and 2nd most predicted classes for each instance, and test whether the difference in their mean predicted probability is statistically significant.

$$\Pr(y = k) = \frac{\exp(-(\mathbf{y}_0 - \mathbf{1}_C)_k^2 / \tau)}{\sum_{i=1}^C \exp(-(\mathbf{y}_0 - \mathbf{1}_C)_i^2 / \tau)}; \hat{y} = \arg \max_k (-(\mathbf{y}_0 - \mathbf{1}_C)_k^2).$$

Classification

Dataset	Accuracy	PIW		Acc. by PIW	Acc. by t -test Result	
		Correct	Incorrect		Rejected	Not-Rejected (Count)
FashionMNIST						
overall	91.79%	0.67	3.20	89.36%	92.07%	55.84% (77)
most acc.	98.50%	0.39	2.08			
least acc.	74.80%	1.37	3.26			
CIFAR-10						
overall	90.95%	2.37	21.52	87.84%	91.25%	42.86% (63)
most acc.	96.00%	0.55	29.27			
least acc.	81.90%	5.48	21.45			
CIFAR-100						
overall	71.42%	0.59	3.91	60.53%	71.56%	35.90% (39)
most acc.	95.00%	0.16	1.92			
least acc.	44.00%	5.09	5.84			
ImageNet-100						
overall	82.34%	2.06	13.73	68.64%	82.90%	34.48% (58)
most acc.	98.00%	0.72	8.06			
least acc.	42.00%	6.79	14.15			
ImageNet (f_ϕ Accuracy 73.87%)						
overall	74.28%	0.65	3.11	69.22%	74.63%	24.93% (349)
most acc.	98.00%	0.27	2.80			
least acc.	8.00%	20.10	50.07			
ImageNet (f_ϕ Accuracy 76.13%)						
overall	76.20%	0.51	3.60	75.21%	76.30%	25.71% (105)
most acc.	98.00%	0.08	2.66			
least acc.	18.00%	1.87	3.26			
ImageNet (f_ϕ Accuracy 80.30%)						
overall	80.35%	1.42	5.13	74.08%	80.59%	27.63% (228)
most acc.	98.00%	0.49	2.34			
least acc.	8.00%	91.70	84.61			

PIW (multiplied by 100), accuracy by predicting with the narrowest PIW, and accuracy by t -test rejection status, for the FashionMNIST, CIFAR-10, CIFAR-100, ImageNet-100, and ImageNet classification tasks, over a single experimental run. We only report the PIW for all test instances, and within the most and least accurate classes.

- Paper: <https://arxiv.org/abs/2206.07275>
- Code: <https://github.com/XzwHan/CARD>