Engression: Extrapolation through the Lens of Distributional Learning

Xinwei Shen

Seminar for Statistics, ETH Zurich

Nicolai Meinshausen

July 14, 2024

Outline

1. A new method called *engression* for distributional learning
2. Applying engression to the extrapolation problem in nonparametric regression

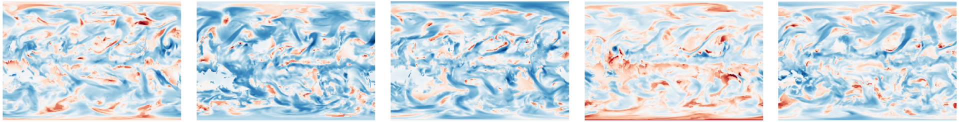
Part I Distributional Learning

Distributional target

Target: the distribution, rather than merely the mean or median

- Climate science: precipitation (mean, variation, extremes, spatial structure, etc)
- Medicine: quantiles of children's height given their age and weight
- ...

Global precipitation fields on different days



Regression

Response $Y \in \mathbb{R}^p$; predictors $X \in \mathbb{R}^d$; training distribution P_{tr}

- L_2 or L_1 regression (Legendre, 1806) for conditional mean or median estimation
- Distributional regression via the cdf (Foresi and Peracchi '95; Hothorn et al. '14), pdf (Dunson et al. '07), or quantiles (Koenker et al. '78; Koenker '05; Meinshausen '06) for conditional distribution estimation

Our target: $P_{\text{tr}}(y|x)$

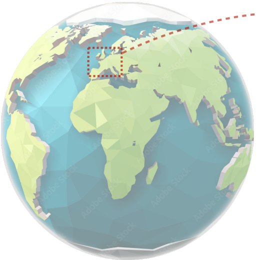
Enough?

Application: climate downscaling

High-dimensional response variables

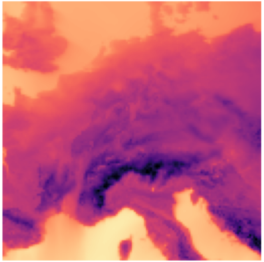
- Physical climate models

Low-resolution



Global climate model (GCM)
X

High-resolution



Regional climate model (RCM)
$Y \in \mathbb{R}^{128 \times 128}$

- Statistical downscaling: emulating RCM by estimating $P_{Y|X}$

Distributional learning via generative modeling

- Build a generative model to describe the target distribution:

$$Y = g(X, \varepsilon)$$

  where $\varepsilon \sim P_\varepsilon$ pre-defined and map $g : (x, \varepsilon) \mapsto y$ is often parametrized by neural networks.
- Rationality: change of variables + universal approximation
- Goal: find g such that $g(x, \varepsilon) \sim P_{\text{tr}}(y|x)$ for any x
- Sampling-based inference: a model to sample from $P_{\text{tr}}(y|x)$.

Our distributional learning method: Engression (S. and Meinshausen, '23)

Model class: $\mathcal{M} = \{g(x, \varepsilon)\}$, where ε is a standard Gaussian. Denote $g(x, \varepsilon) \sim P_g(y|x)$.

Engression: **En**ergy score re**gression**

$$\tilde{g} \in \operatorname*{argmin}_{g \in \mathcal{M}} \mathbb{E}_{(X,Y)\sim P_{\text{tr}}}[-\text{ES}(P_g(y|X), Y)]$$

Energy score (Gneiting and Raftery, '07)

Definition. Given a distribution P and an observation z, the energy score is defined as

$$\text{ES}(P, z) = \frac{1}{2}\mathbb{E}_{(Z,Z')\sim P\otimes P}\|Z - Z'\|_2 - \mathbb{E}_P\|Z - z\|_2.$$

Lemma. For any P, we have $\mathbb{E}_{Z\sim P^\star}[\text{ES}(P, Z)] \le \mathbb{E}_{Z\sim P^\star}[\text{ES}(P^\star, Z)]$, where "=" $\Leftrightarrow P = P^\star$.

Corollary. Under correct model specification, we have $\tilde{g}(x, \varepsilon) \sim P_{\text{tr}}(y|x), \forall x \in \text{supp}(P_{\text{tr}}(x))$.

Engression (explicitly):

$$\min_{g \in \mathcal{M}} \mathbb{E}\Big[\|Y - g(X,\varepsilon)\|_2 - \frac{1}{2}\|g(X,\varepsilon) - g(X,\varepsilon')\|_2\Big]$$

- Parametrized by neural networks
- Optimized by gradient-based algorithms

Point estimation by Monte Carlo: for fixed x, draw samples of ε

- Conditional mean estimation: $\hat{\mathbb{E}}_\varepsilon[\tilde{g}(x,\varepsilon)]$
- Conditional α-quantile estimation: $\hat{Q}_\alpha(\tilde{g}(x,\varepsilon))$

Our R and Python packages (`http://github.com/xwshen51/engression`)

R: `install.packages("engression")`

Python: `pip install engression`

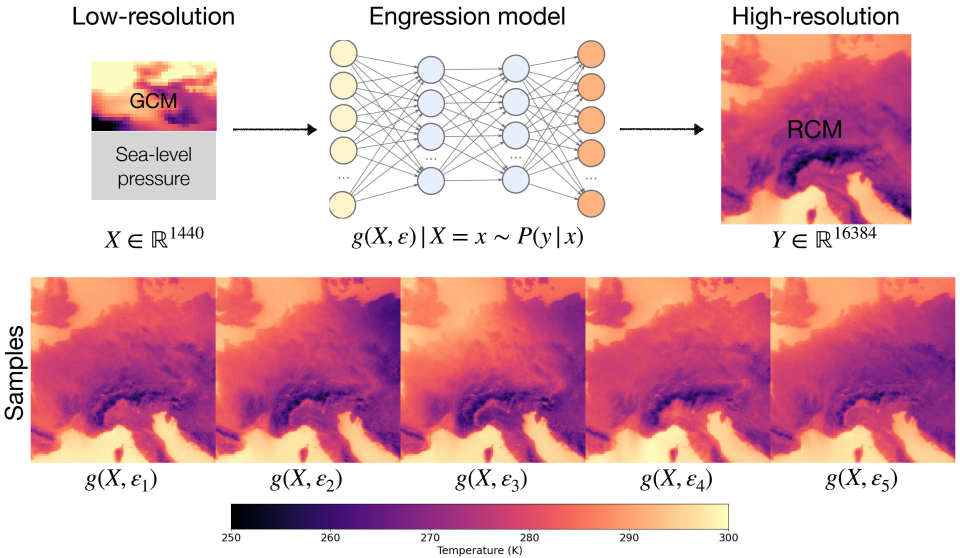
Support general data types and tasks:

- X, Y can be multivariate; continuous or categorical
- Estimation for the conditional mean or quantiles
- Sampling from the estimated distribution

Demo:

```
> library(engression)                                                  ## load engression package
> engressionFit = engression(X, Y)                                     ## fit an engression model
> predict(engressionFit, Xtest, type="mean")                           ## mean prediction
> predict(engressionFit, Xtest, type="quantile", quantiles=c(0.1, 0.5, 0.9)) ## quantile prediction
> predict(engressionFit, Xtest, type="sample", nsample=100)            ## sampling
```

Engression for downscaling (Joint with Maybritt Schillinger, Maxim Samarin, and Nicolai Meinshausen)



Summary of Part I

Engression as a general distributional learning method

- Estimate (conditional) distributions
- Compared to traditional distributional regression (e.g., quantile regression):
  - no quantile crossing
  - expressive capacity of neural networks alleviates limitations of parametric model specifications
  - scalable to (very) high-dimensional X and Y
- Compared to modern generative models (e.g., diffusion model, GAN):
  - computationally lighter, fewer tuning parameters, especially suitable for non-image data

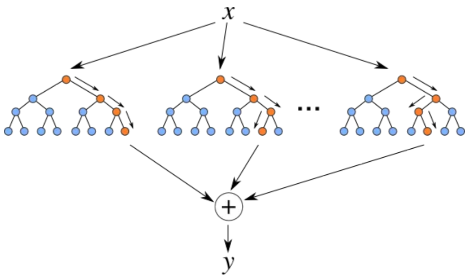
Part II Extrapolation in Nonparametric Regression

Today's prediction models

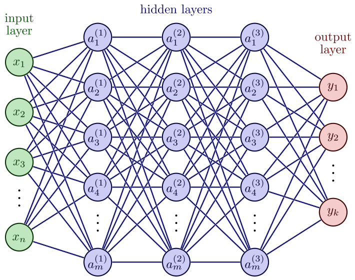
Linear models

$$Y = \beta^\top X + \varepsilon$$

Random Forests, gradient-boosted trees



Neural networks



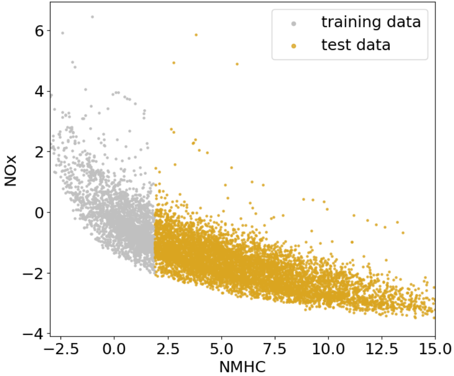
What could go wrong?

It is common to observe training data within a bounded support and encounter test data outside the training support.

- Biodiversity: predicting how species respond to climate change
- Counterfactual prediction: covariate shifts from the treatment to control groups
- ...

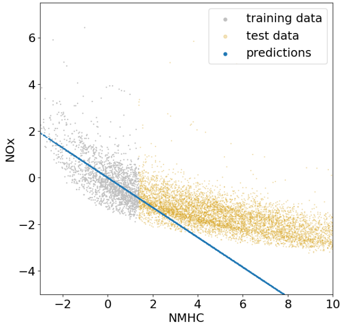
Extrapolation is a fundamental challenge for nonlinear regression.

Air quality data example

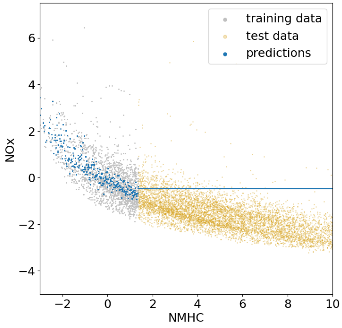


Measurements of two pollutants: Total Nitrogen Oxides (NOx) and non-methane hydrocarbons (NMHC) concentration.

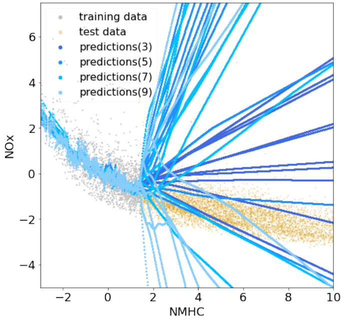
Challenge of nonlinear extrapolation



Linear regression



Random Forests

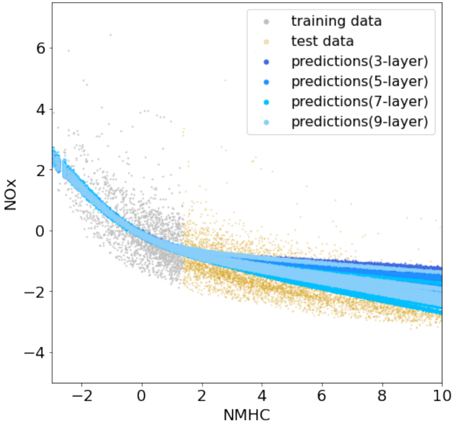


Neural network regression[1]

[1] Predictions from different random initializations and NN architectures with 3, 5, 7, or 9 layers

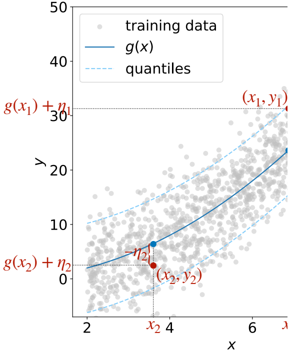
Engression makes a difference

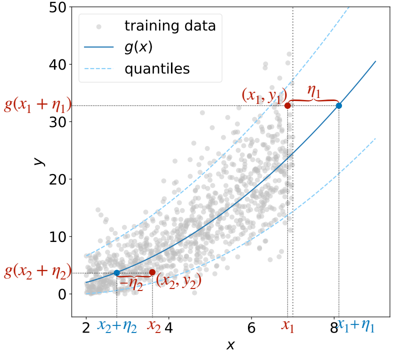
The reliability of engression does not break down immediately at the support boundary.



Results of engression with 3, 5, 7, or 9 layers and random initializations.

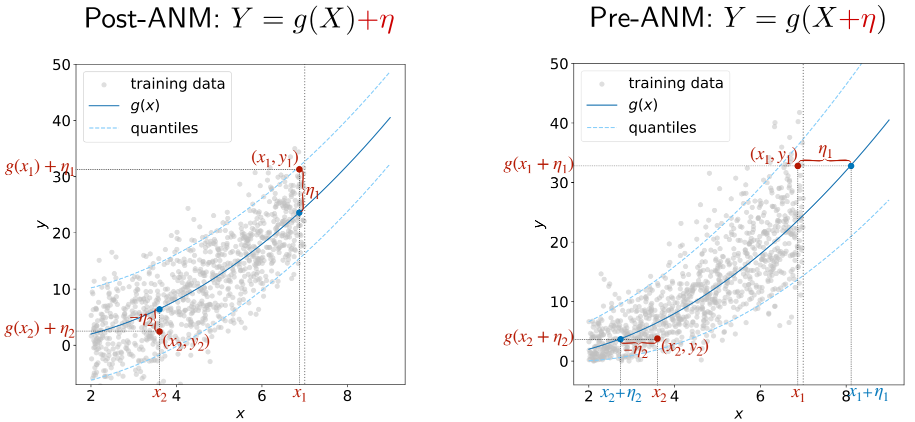
Additive noise models (ANMs)





All models are wrong, but can one of them be useful in terms of extrapolation?

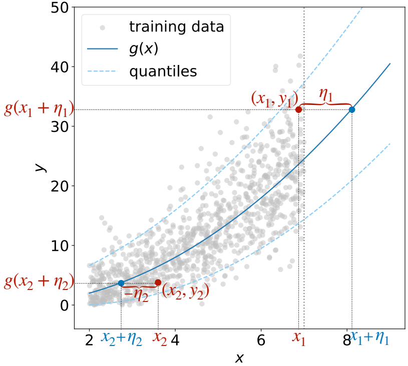
Additive noise models (ANMs)



💡 Pre-additive noises reveal some information about the true function outside the support.

Distributional learning

Pre-ANM: $Y = g(X + \eta)$



To capture the information from the pre-additive noise, one needs to fit the full conditional distribution of Y given X.

Engression has the two ingredients for extrapolation

✓ Engression is **a distributional learning method**.

✓ Engression model $\mathcal{M} = \{\mathbf{g}(x, \varepsilon)\}$ contains **pre-ANMs** $\{g(W^\top x + h(\varepsilon)) : g \in \mathcal{G}, h \in \mathcal{H}\}$, where $h(\varepsilon)$ represents the pre-additive noise; g, h, and W are to be learned.

Regression fails to extrapolate

Setup:

- True model $Y = g^\star(X+\eta)$; pre-ANM class $\mathcal{M} = \{g(x+h(\varepsilon)) : g \in \mathcal{G}, h \in \mathcal{H}\}$; $\mathcal{G}$ strictly monotone;
- (For simplicity) symmetric noise $\eta \in [-\eta_{\max}, \eta_{\max}]$; training support $(-\infty, x_{\max}]$.

Proposition (S. and Meinshausen, '23)

Let $\mathcal{F}_{L_1} := \operatorname{argmin}_{g\in\mathcal{G}} \mathbb{E}_{P_{\mathrm{tr}}}|Y - g(X)|$. For any $x > x_{\max}$, we have

$$\sup_{g\in\mathcal{F}_{L_1}} |g(x) - g^\star(x)| = \infty.$$

Engression can extrapolate up to a certain point

Setup:

- True model $Y = g^\star(X + \eta)$; pre-ANM class $\mathcal{M} = \{g(x + h(\varepsilon)) : g \in \mathcal{G}, h \in \mathcal{H}\}$; $\mathcal{G}$ strictly monotone;
- (For simplicity) symmetric noise $\eta \in [-\eta_{\max}, \eta_{\max}]$; training support $(-\infty, x_{\max}]$.

Theorem (S. and Meinshausen, '23)

We have $\tilde{g}(x) = g^\star(x)$ for all $x \leq x_{\max} + \eta_{\max}$, and $\tilde{h}(\varepsilon) \overset{d}{=} \eta$.

- Population engression $(\tilde{g}, \tilde{h})$ recovers the true model beyond the training support.
- Blessing of noise: the more (pre-additive) noise there is, the farther one can extrapolate.

Relax the assumptions?

"truth $Y = g^\star(X + \eta)$; pre-ANM class $\mathcal{M} = \{g(x + h(\varepsilon)) : g \in \mathcal{G}, h \in \mathcal{H}\}$; $\mathcal{G}$ monotone"?

- Model $Y = g^\star(X + \eta) + \xi$ to allow both pre and post-additive noises
- Monotone $g^\star$ only around the support boundary.

For conditional distribution estimation, engression is rather general.

In practice, engression uses general models $\{g(x, \varepsilon)\}$.

Finite-sample bounds for quadratic models

Setup:

- Quadratic pre-ANM class:

$$\{\beta_0 + \beta_1(x+\eta) + \beta_2(x+\eta)^2 : \beta = (\beta_0, \beta_1, \beta_2) \in \mathcal{B}, \eta \sim P_\eta \in \mathcal{P}_\eta\},$$

- Training support $\mathcal{X} = \{x_1, x_2\}$
- Training data: $(x_1, Y_{1,i}), i = 1, \ldots, n$ and $(x_2, Y_{2,i}), i = 1, \ldots, n$

Failure of L_2 and quantile regression

L_2 regression estimators:

$$\mathcal{B}^{\mu} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2n} \sum_{j=1}^{2} \sum_{i=1}^{n} [Y_{j,i} - (\beta_0 + \beta_1 x_j + \beta_2 x_j^2)]^2$$

Quantile regression estimators:

$$\mathcal{B}^{q}_{\alpha} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2n} \sum_{j=1}^{2} \sum_{i=1}^{n} \rho_{\alpha}(Y_{j,i} - (\beta_0 + \beta_1 x_j + \beta_2 x_j^2))$$

Proposition

For all $x \notin \mathcal{X}$, we have

$$\sup_{\beta \in \mathcal{B}^{\mu}} \mathbb{E}[(Y - (\beta_0 + \beta_1 x + \beta_2 x^2))^2] = \infty,$$

$$\sup_{\beta \in \mathcal{B}^{q}_{\alpha}} |(q^{\star}_{\alpha}(x) - (\beta_0 + \beta_1 x + \beta_2 x^2)| = \infty.$$

Finite-sample bounds for engression

Theorem

With probability exceeding $1-\delta$, we have

$$\|\hat{\beta}-\beta^{\star}\| \leq \frac{C_1}{(x_2-x_1)}\left(\frac{\log(2/\delta)}{n}\right)^{\frac{1}{3}}.$$

For any $x \in \mathbb{R}$, it holds with probability exceeding $1-\delta$ that

$$(\hat{\mu}(x)-\mu^{\star}(x))^2 \leq C_2 \max\{1,|x|,x^2\}\left(\frac{\log(2/\delta)}{n}\right)^{\frac{2}{3}}.$$

For any $x \in \mathbb{R}$ and $\alpha \in [0,1]$, it holds with probability exceeding $1-\delta$ that

$$|\hat{q}_\alpha(x)-q^{\star}_\alpha(x)| \leq C_3 \max\{1,|x|,x^2\}|Q^{\eta^\star}_\alpha|\left(\frac{\log(2/\delta)}{n}\right)^{\frac{1}{3}}.$$

Misspecified pre-ANM

True data generating model is a post-ANM: $Y = \beta_0^\star + \beta_1^\star x + \beta_2^\star x^2 + \eta^\star$
With a quadratic pre-ANM class.

Proposition

With probability exceeding $1 - \delta$, we have

$$\max\left\{|\hat{\beta}_0 - (\beta_0^\star - \beta_2^\star x_1 x_2)|,\ |\hat{\beta}_1 - (\beta_1^\star + \beta_2^\star(x_1 + x_2))|,\ |\hat{\beta}_2|\right\} \lesssim \left(\frac{\log(2/\delta)}{n}\right)^{\frac{1}{3}}.$$

Defaults to a linear extrapolation ✓

Consistency for general pre-ANMs

General pre-ANM class:

$$\{g(x + h(\varepsilon)) : g \in \mathcal{G}, h \in \mathcal{H}\}$$

Training support $\mathcal{X} = [x_{\min}, x_{\max}]$
True model $g^\star(x + h^\star(\varepsilon))$

Theorem (S. and Meinshausen, '23)

Under suitable conditions, we have for all $\tilde{x} \in \tilde{\mathcal{X}} := \{x + h^\star(\varepsilon) : x \in \mathcal{X}, \varepsilon \in [0,1]\}$ and $\varepsilon \in [0,1]$

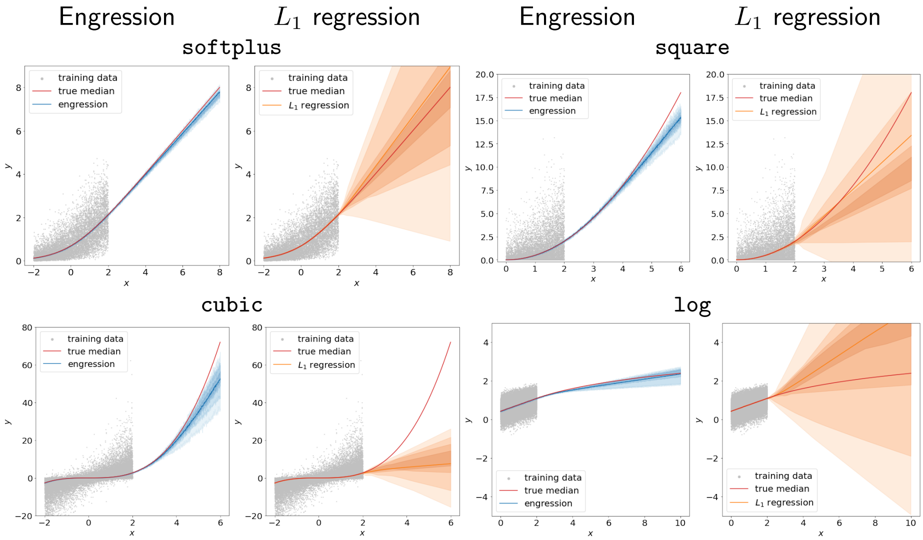
$$\hat{g}(\tilde{x}) \xrightarrow{p} g^\star(\tilde{x}) \quad \text{and} \quad \hat{h}(\varepsilon) \xrightarrow{p} h^\star(\varepsilon) \quad \text{as } n \to \infty.$$

Simulation settings

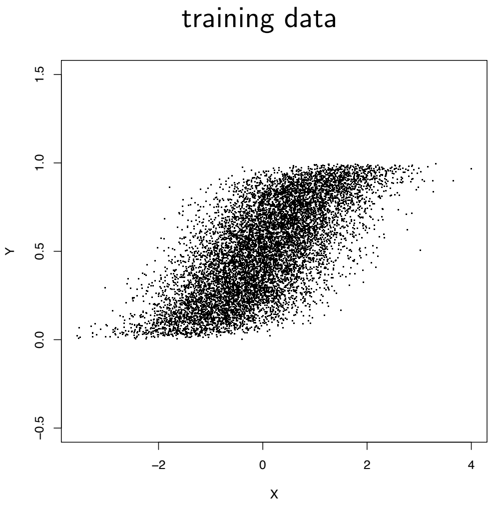
Table: $Y = g^\star(X+\eta)$, $x_{\max} = 2$, $\eta_{\max} \approx 2$

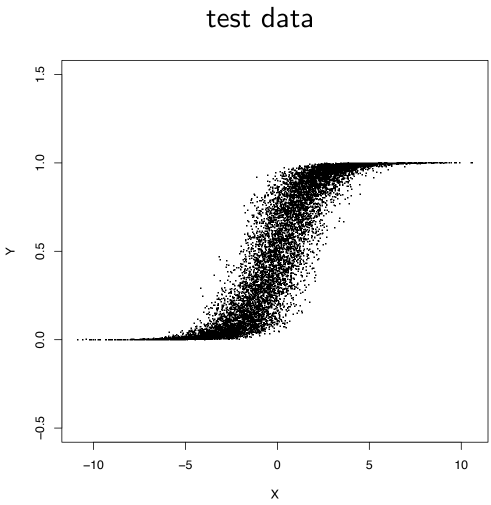
Name	$g^\star(\cdot)$	X	η
`softplus`	$g^\star(x) = \log(1+e^x)$	Unif$[-2,2]$	$\mathcal{N}(0,1)$
`square`	$g^\star(x) = (x_+)^2/2$	Unif$[0,2]$	$\mathcal{N}(0,1)$
`cubic`	$g^\star(x) = x^3/3$	Unif$[-2,2]$	$\mathcal{N}(0,1.1^2)$
`log`	$g^\star(x) = \begin{cases} \frac{x-2}{3} + \log(3) & x \leq 2 \\ \log(x) & x > 2 \end{cases}$	Unif$[0,2]$	$\mathcal{N}(0,1)$

Conditional median estimation

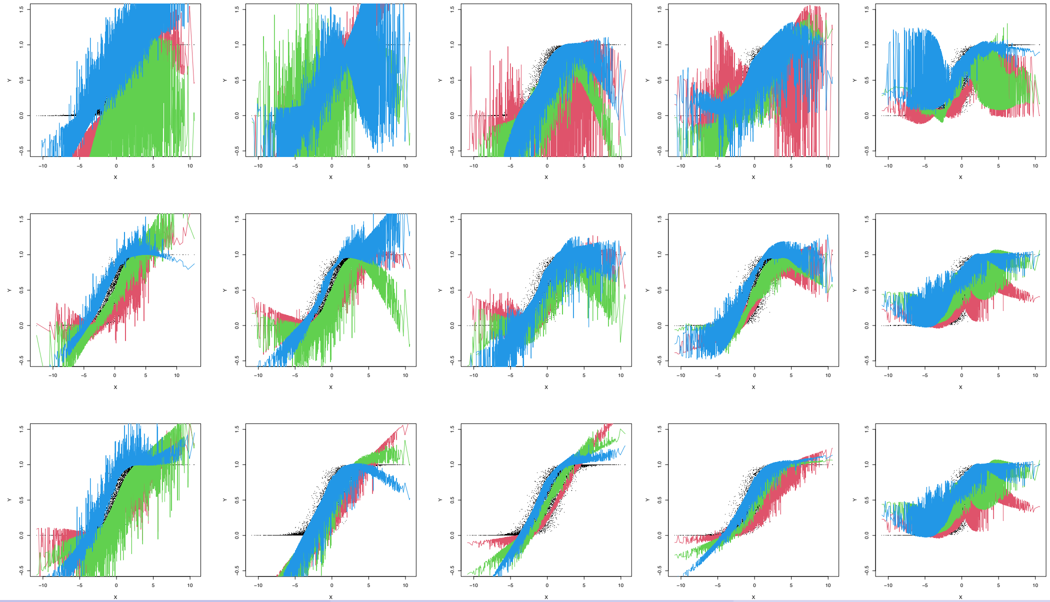


Numerical example

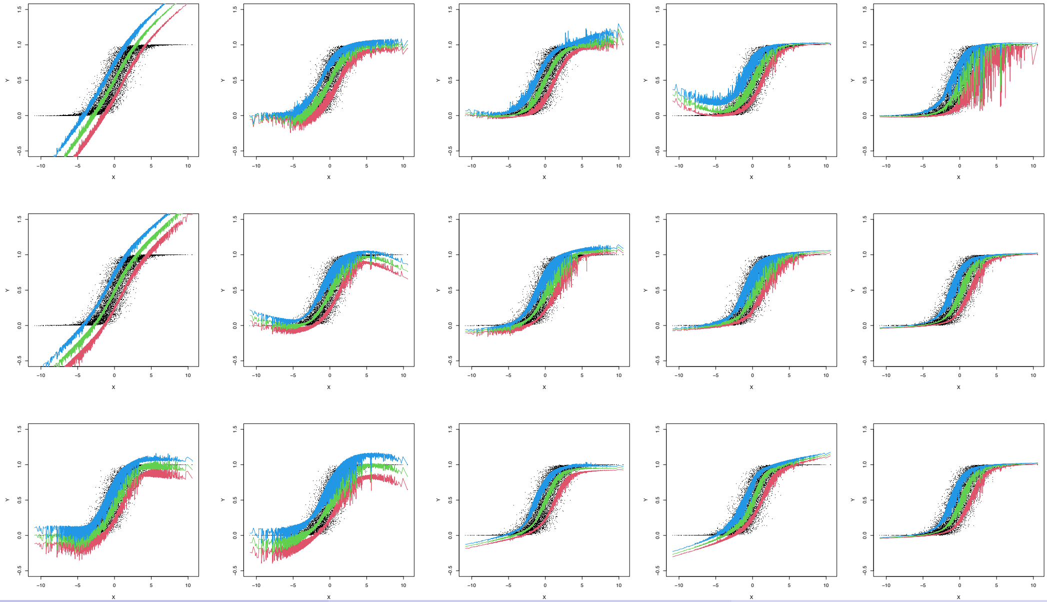




NN quantile regression. Top to bottom: $10, 100$ and 1000 hidden dimension. Left to right: $2, 3, 5, 10$ and 20 layers.



Engression. Top to bottom: $10, 100$ and 1000 hidden dimension. Left to right: $2, 3, 5, 10$ and 20 layers.



Large-scale real-data experiments for univariate prediction

590 data configurations:

- *Real data sets* from various application domains
- *Pairwise prediction* for all variables
- *Split the training and test data* at the 0.3–0.7 quantiles of the predictor

18 hyperparameter settings of neural network architectures and optimization

In total: 590*18=10'620 models for each method

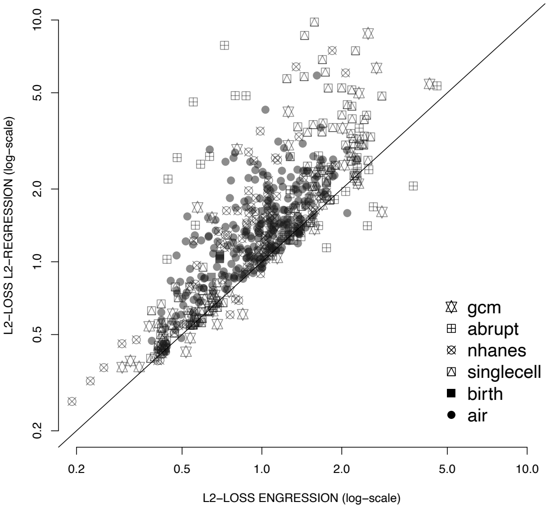
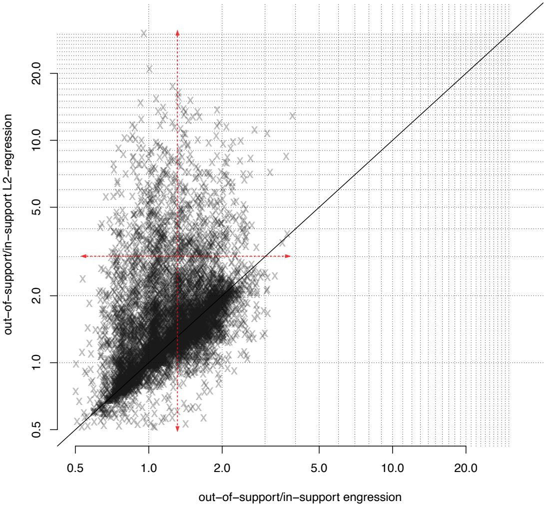


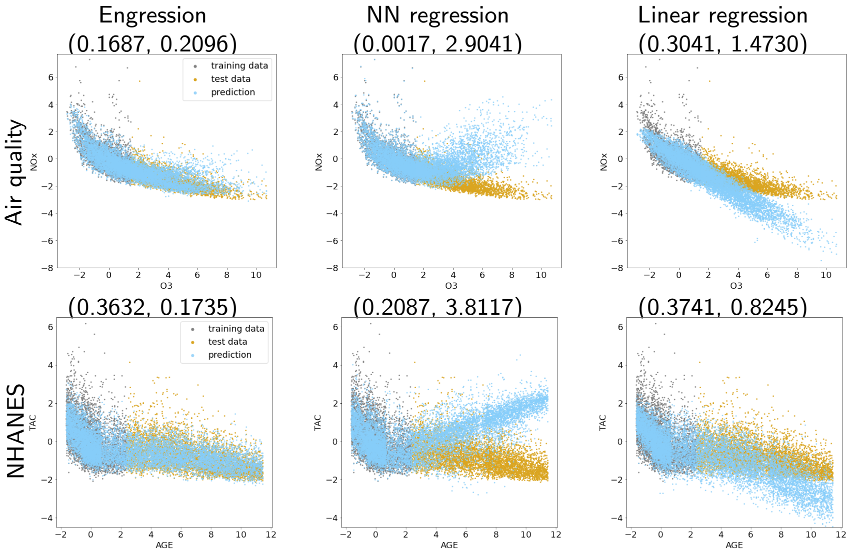
Figure: Out-of-support losses (in log-scale) of engression and regression for various data configurations, averaging over all hyperparameter settings.

The ratio (in log-scale) between out-of-support and in-support L_2 losses of engression and regression for all hyperparameter settings.



- Engression has comparable out-of-support and in-support performance.
- Regression degrades drastically out-of-support.
- Engression is much more robust to the choice of hyperparameters than NN regression.

Multivariate prediction



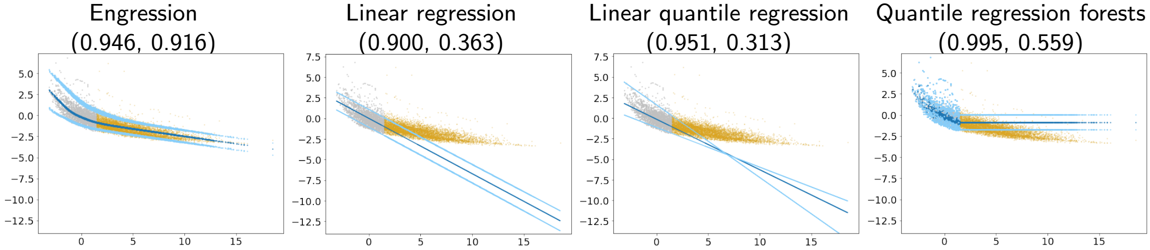
Prediction intervals

Proposition (S. and Meinshausen, '23)

For $\alpha \in [0,1]$, it holds for all $x \le x_{\max} + \eta_{\max} - Q_\alpha(\eta)$ that $\tilde{q}_\alpha(x) = q^\star_\alpha(x)$, i.e.,

$$\mathbb{P}\left(Y \le \tilde{q}_{1-\alpha}(X) \mid X = x\right) = 1 - \alpha.$$

$\Rightarrow$ prediction intervals with conditional coverage guarantee outside the support (in population).



Summary of Part II

Engression for extrapolation

- Inferential target: conditional mean or quantile function beyond the training support
- Recipe: distributional learning + pre-additive noise models

Outlook

- For statisticians, engression provides a flexible tool for statistical inference problems that involve distribution estimation.
- For applied researchers, engression can be an interesting addition to the current data analysis toolkit: comprehensive quantification of the full distribution; different behavior when it comes to data outside the training support

Outlook

- Robustness (invariance) against distribution shifts:
  Henzi, S., Law, and Bühlmann. Invariant Probabilistic Prediction. arXiv:2309.10083
- Dimensionality reduction (unsupervised):
  S. and Meinshausen. Distributional Principal Autoencoders. arXiv:2404.13649
- Distributional causal effect estimation: coming soon