Engression for Distributional Learning and Extrapolation

Xinwei Shen

Seminar for Statistics, ETH Zurich

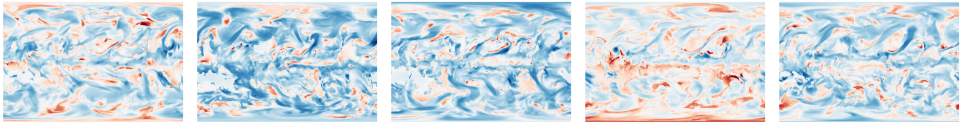
Joint with Nicolai Meinshausen

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: $P_{\mathrm{tr}}(y|x)$

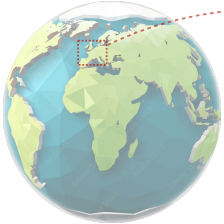
Enough?

Application: climate downscaling

High-dimensional response variables

- Physical climate models

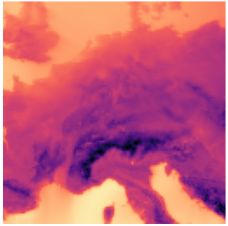
Low-resolution



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 $g(x, \varepsilon) \sim P_{\mathrm{tr}}(y|x)$ for any x
- Sampling-based inference: a model to sample from $P_{\mathrm{tr}}(y|x)$.

Our distributional learning method: Engression (S. and Meinshausen, '23)[1]

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**

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

Energy score (Gneiting and Raftery, '07)

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

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

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

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

[1] arXiv:2307.00835

Engression (S. and Meinshausen, '23)

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`

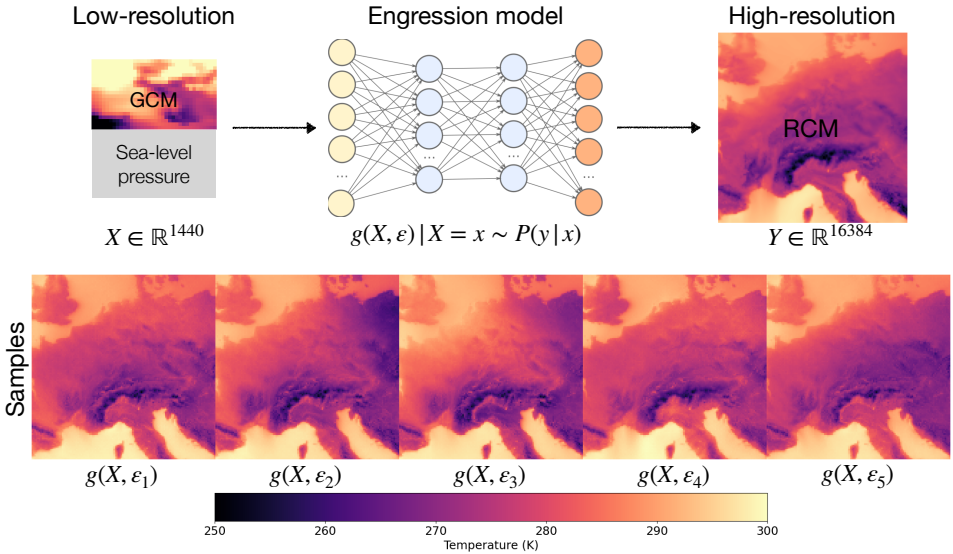
Support general data types and tasks:

- 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, Sebastian Sippel, and Nicolai Meinshausen)



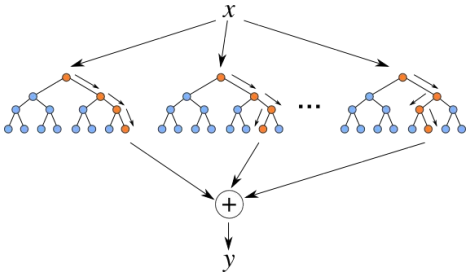
Extrapolation in Nonlinear Regression

Today's prediction models

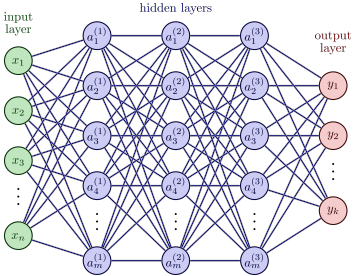
Linear models

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

Random Forests, gradient-boosted trees



Neural networks



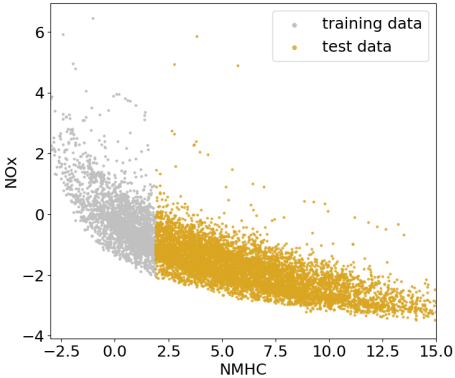
What could go wrong?

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
- ...

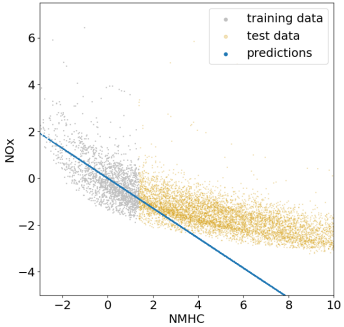
Extrapolation is a fundamental challenge for nonlinear regression.

Air quality data example

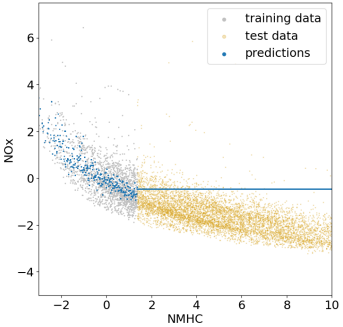


Measurements of two pollutants: Total Nitrogen Oxides (NOx) and non-methane hydrocarbons (NMHC) concentration.

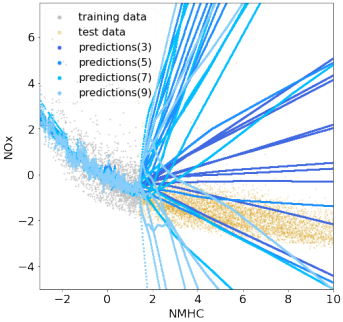
Challenge of nonlinear extrapolation



Linear regression



Random Forests

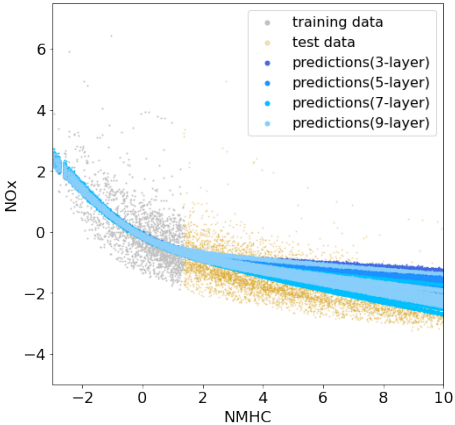


Neural network regression[1]

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

Engression makes a difference

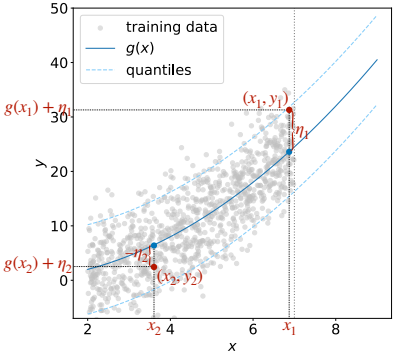
The reliability of engression does not break down immediately at the support boundary.



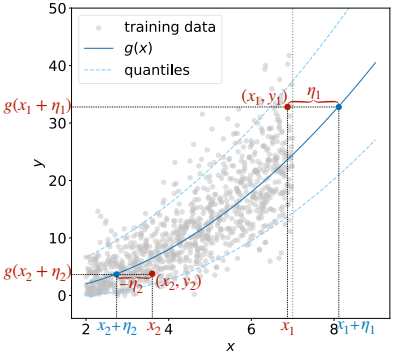
Results of engression with 3, 5, 7, or 9 layers and random initializations.

Additive noise models (ANMs)

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



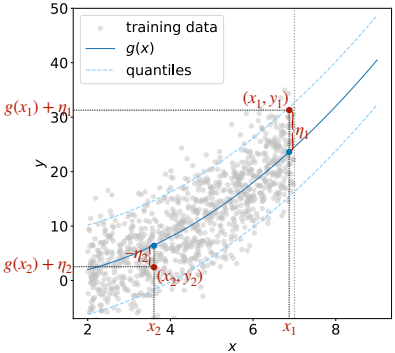
Pre-ANM: $Y = g(X + \eta)$



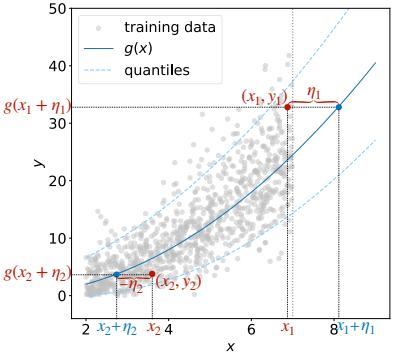
All models are wrong, but can one of them be useful in terms of extrapolation?

Additive noise models (ANMs)

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



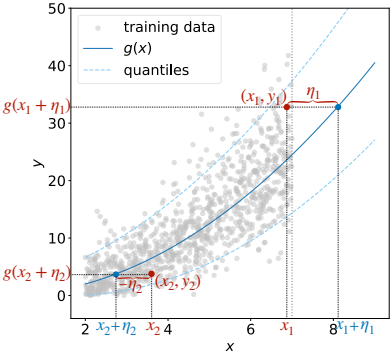
Pre-ANM: $Y = g(X + \eta)$



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 $\mathcal{M} = \{g(x, \varepsilon)\}$ 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; both g and h 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 $\tilde{h}(\varepsilon) \stackrel{d}{=} \eta$.

- 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.

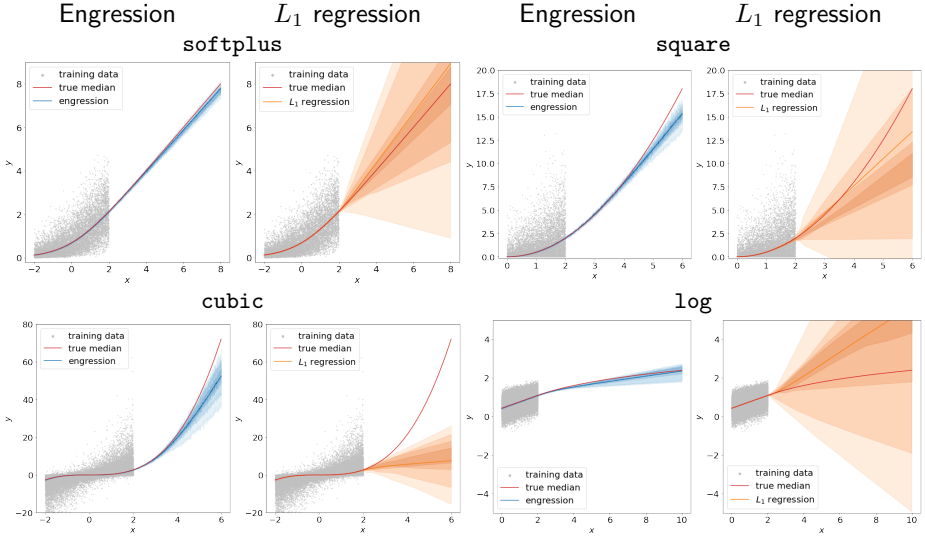
In practice, engression uses general models $\{g(x, \varepsilon)\}$.

Simulation settings

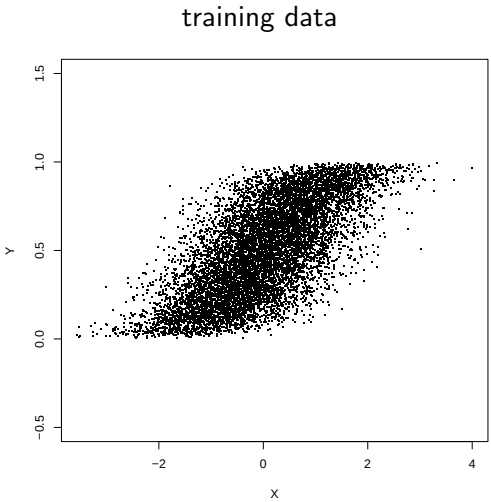
Table: $Y = g^\star(X + \eta)$, $x_{\max} = 2$, $\eta_{\max} \approx 2$

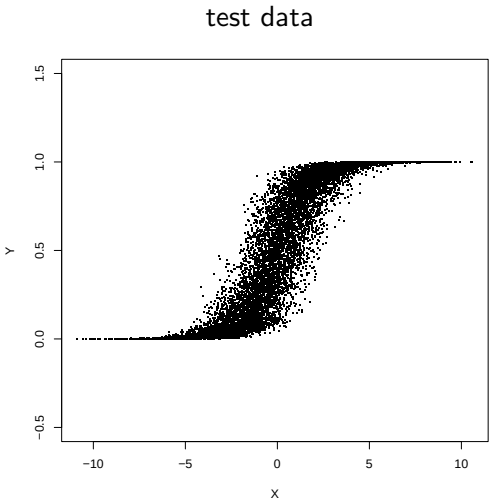
Name	$g^\star(\cdot)$	X	η
`softplus`	$g^\star(x) = \log(1 + e^x)$	Unif$[-2, 2]$	$\mathcal{N}(0, 1)$
`square`	$g^\star(x) = (x_+)^2/2$	Unif$[0, 2]$	$\mathcal{N}(0, 1)$
`cubic`	$g^\star(x) = x^3/3$	Unif$[-2, 2]$	$\mathcal{N}(0, 1.1^2)$
`log`	$g^\star(x) = \begin{cases} \frac{x-2}{3} + \log(3) & x \le 2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

Conditional median estimation

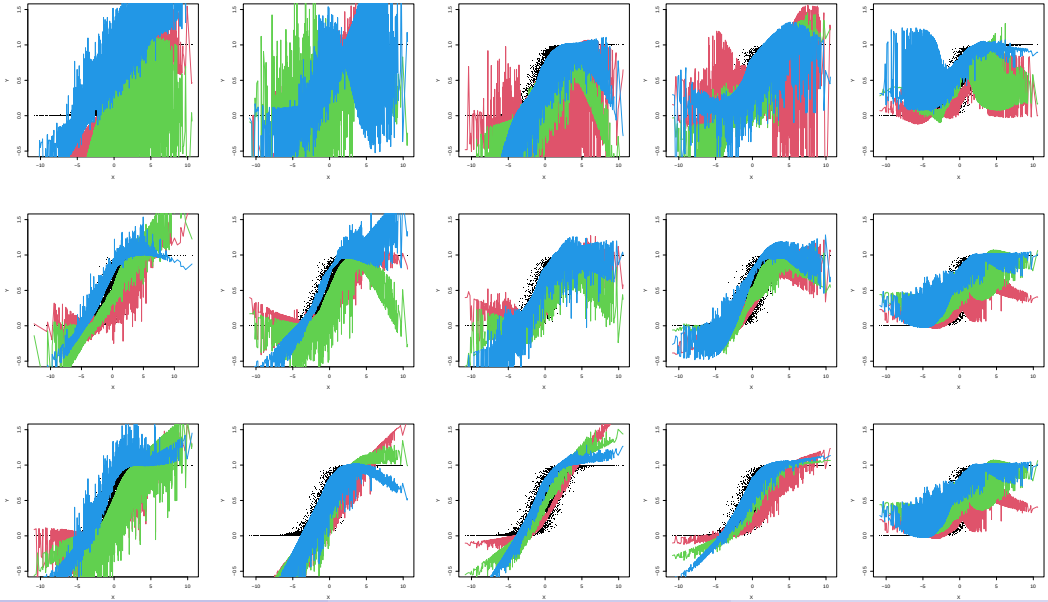


Numerical example

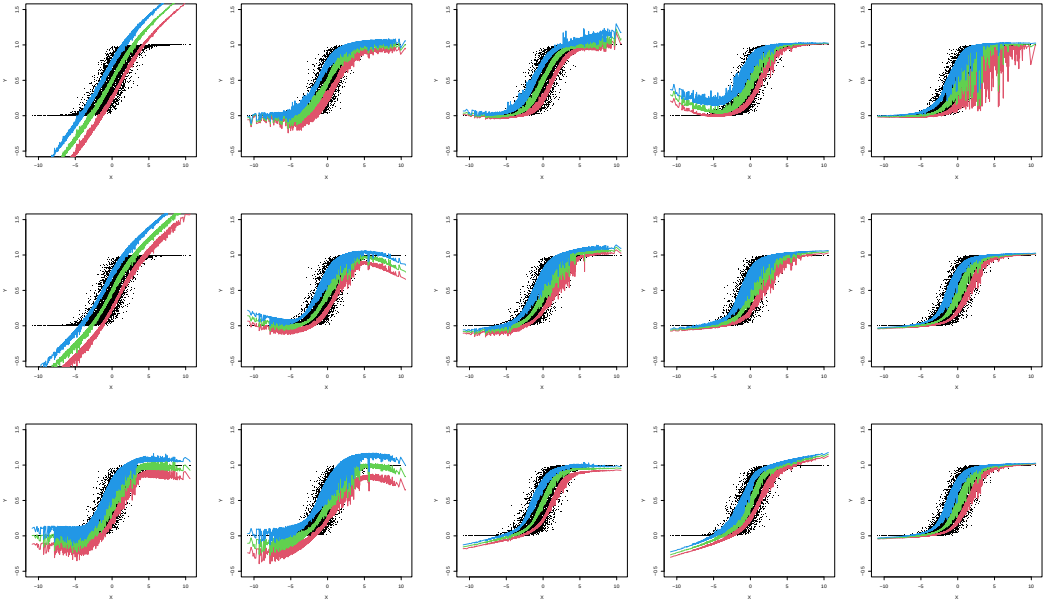




NN quantile regression. Top to bottom: $10, 100$ and 1000 hidden dimension. Left to right: $2, 3, 5, 10$ and 20 layers.



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

- Report the average performance
- Same for engression and regression

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

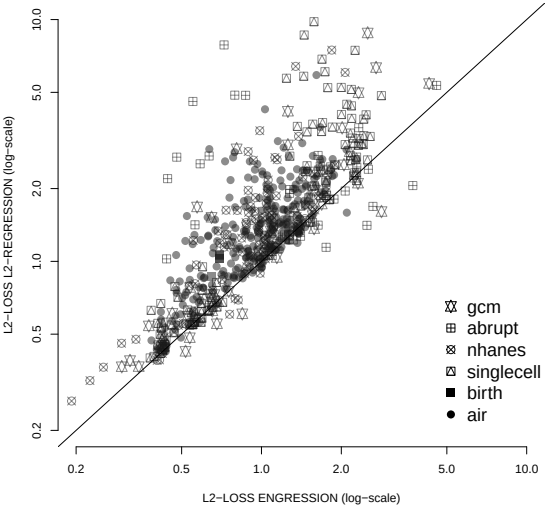
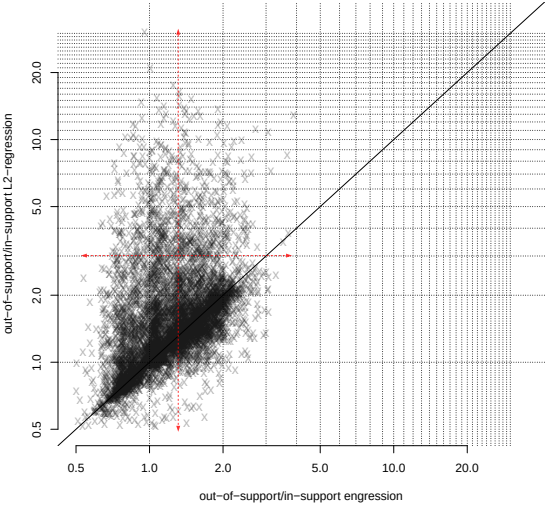


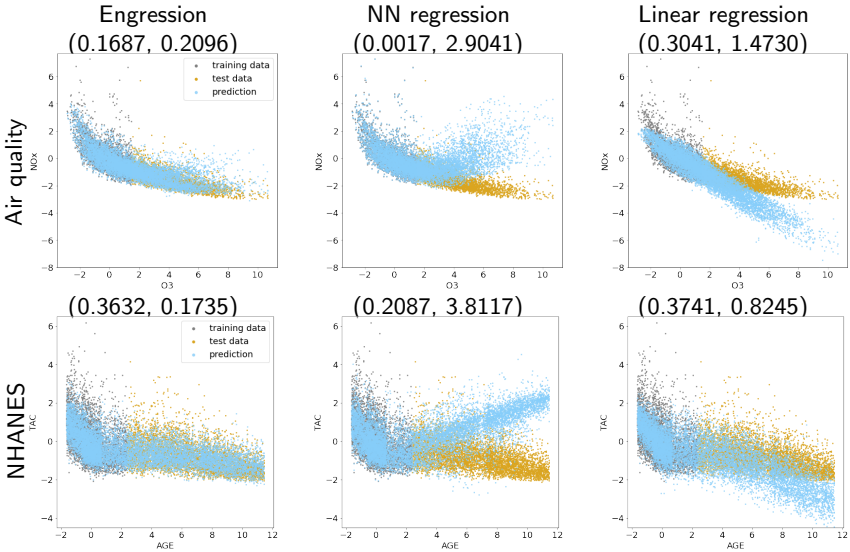
Figure: Out-of-support losses (in log-scale) of engression and regression for various data configurations, averaging over all hyperparameter settings.

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



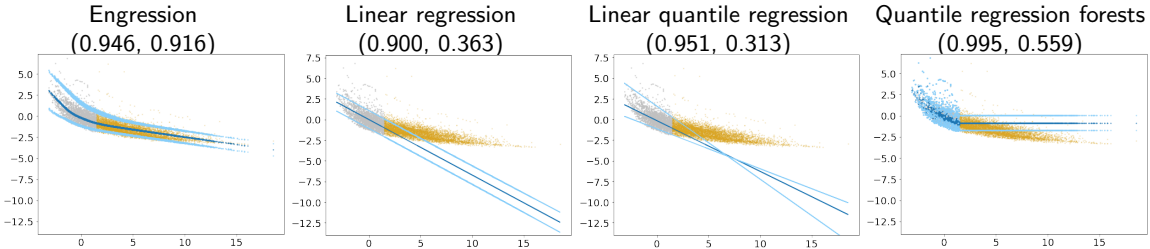
Prediction intervals

Proposition (S. and Meinshausen, '23)

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

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

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



Summary

Engression is a nonlinear distributional regression method.

- For (conditional) distribution estimation:
  - Compared to classical distributional regression: more adaptable to high-dimensional X and Y
  - Compared to generative models: computationally lighter, fewer tuning parameters
- For standard regression:
  - Extrapolation: more reliable for out-of-support test data
  - More robust to different hyperparameters than NN regression (less tuning)
  - Various regression tasks (mean or quantile prediction, prediction interval, sampling)

arXiv:2307.00835

Outlook

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