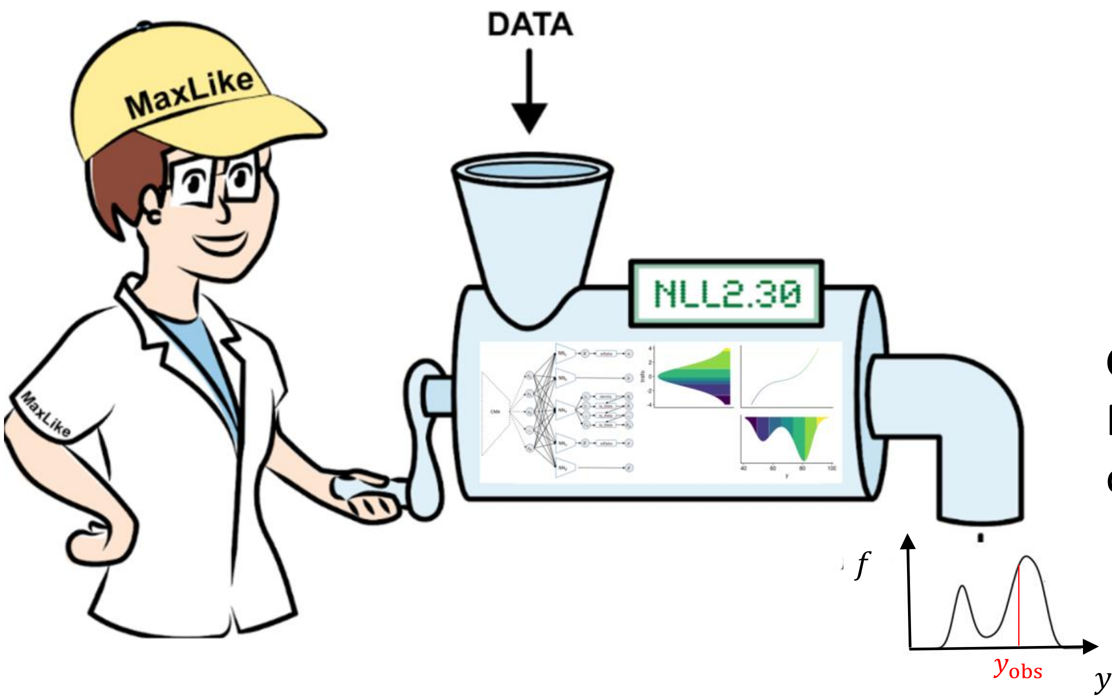
Deep transformation models
for tackling complex probabilistic regression problems

https://arxiv.org/abs/2004.00464



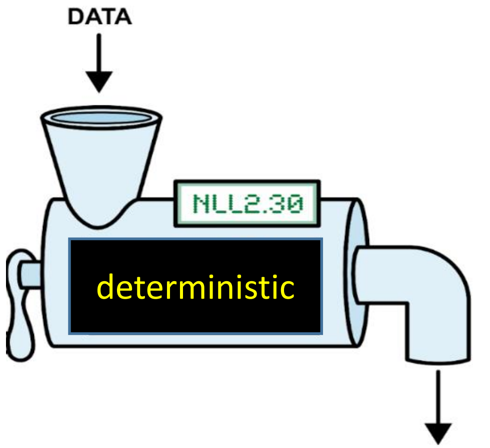
Goal:
Predict a flexible outcome distribution

Beate Sick *
EBPI, University of Zurich &
IDP, Zurich University of Applied Sciences
Email: beate.sick@uzh.ch, sick@zhaw.ch

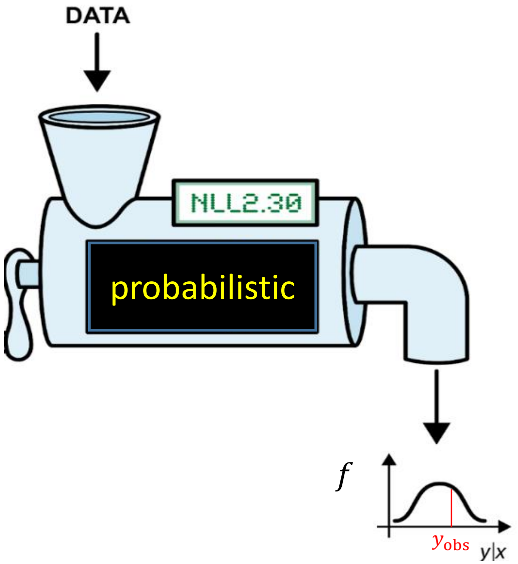
Torsten Hothorn
EBPI, University of Zurich
Email: torsten.hothorn@uzh.ch

Oliver Dürr *
IOS, Konstanz University of Applied Sciences
Email: oliver.duerr@htwg-konstanz.de

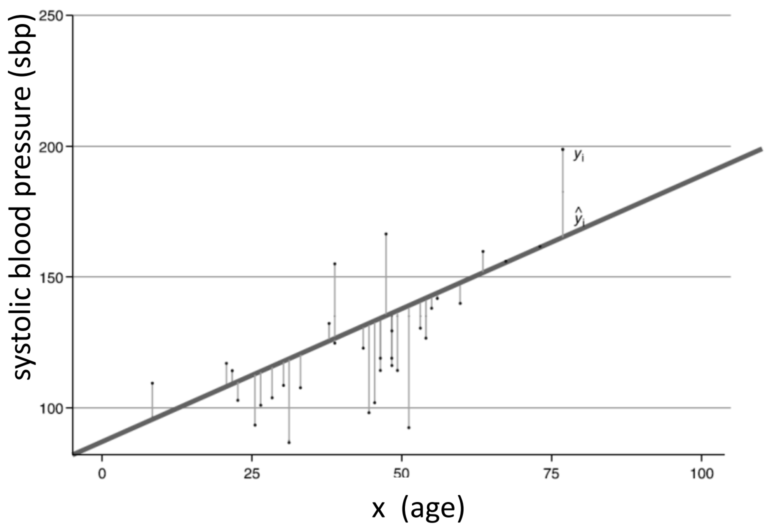
Non-probabilistic versus probabilistic regression DL models

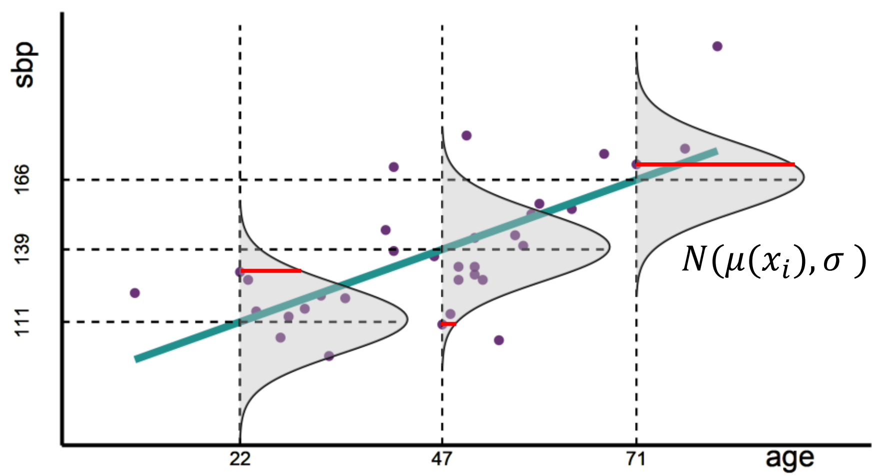


$\hat{y} = 117$



$$\theta(x) = \operatorname{argmin}_{\mu} \sum_{i=1} \log(N(y_i; \mu(x_i), \sigma))$$





Have a look on more complex conditional probability distributions

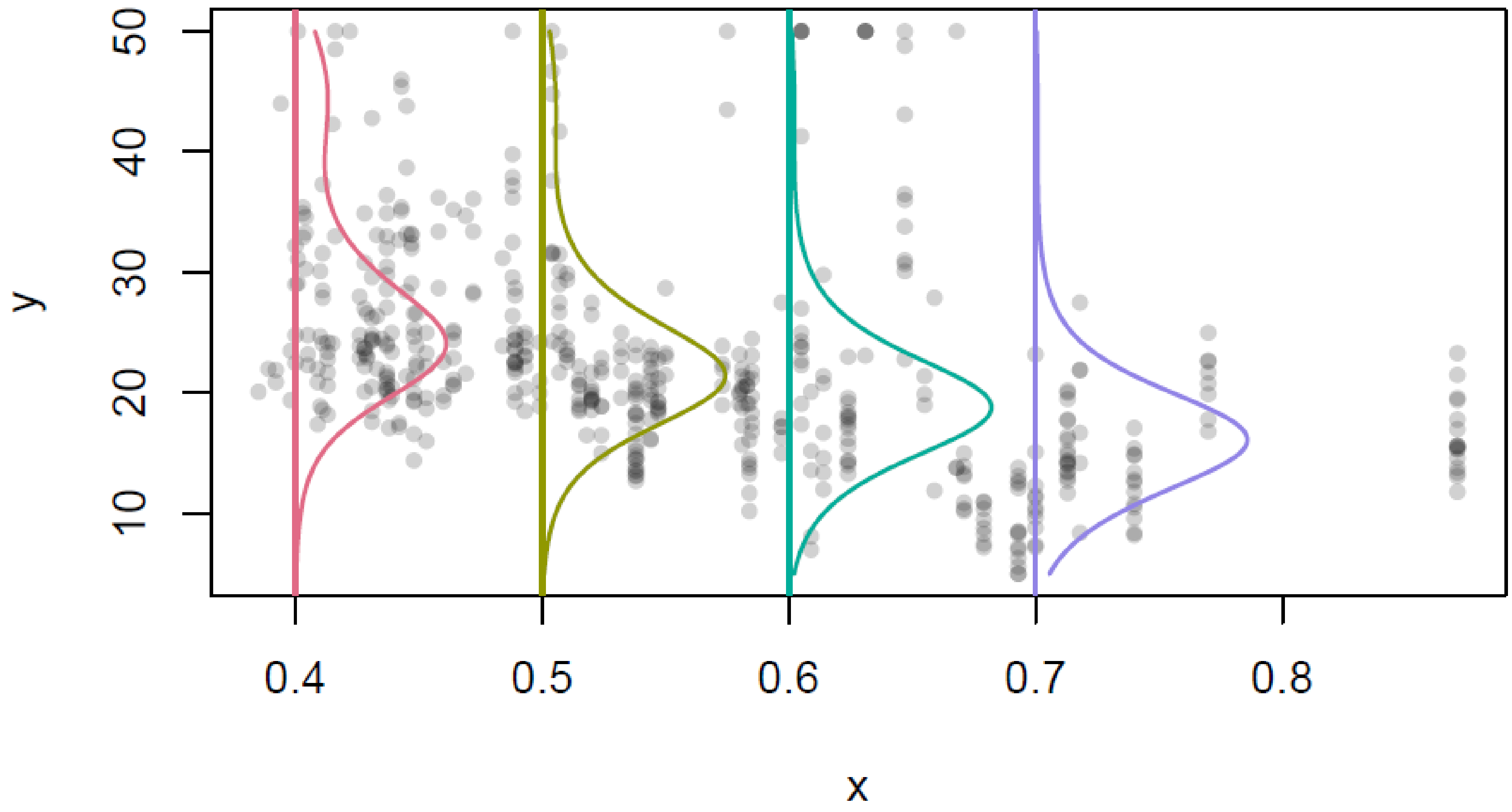


Image credits: Lucas Kook

How to model complex distributions?

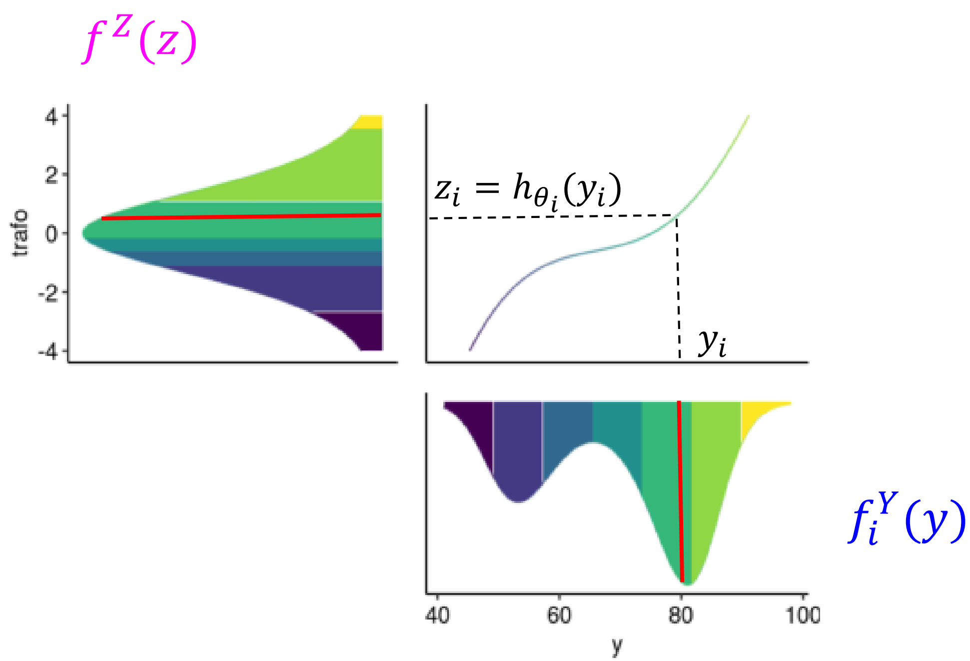
- Use a mixture model (e.g. mixture Gaussians)
- Use a transformation model!

Idea of a transformation model

Idea: we fit a transformation function h that transforms the flexible conditional outcome $f_i^Y(y)$ to an easy (here $N(0,1)$) distribution $f^Z(z)$

$$\mathrm{NLL} = \sum_i -\log\left(f_i^y(y_i)\right) = \sum_i -\log\left(f^z(z_i) \cdot \left|\frac{\partial h_{\hat{\theta}_i}}{\partial y}\right|\Big|_{y_i}\right)$$

"change of variable" formula



For non-Gaussian CPDs we need a non-linear transformation

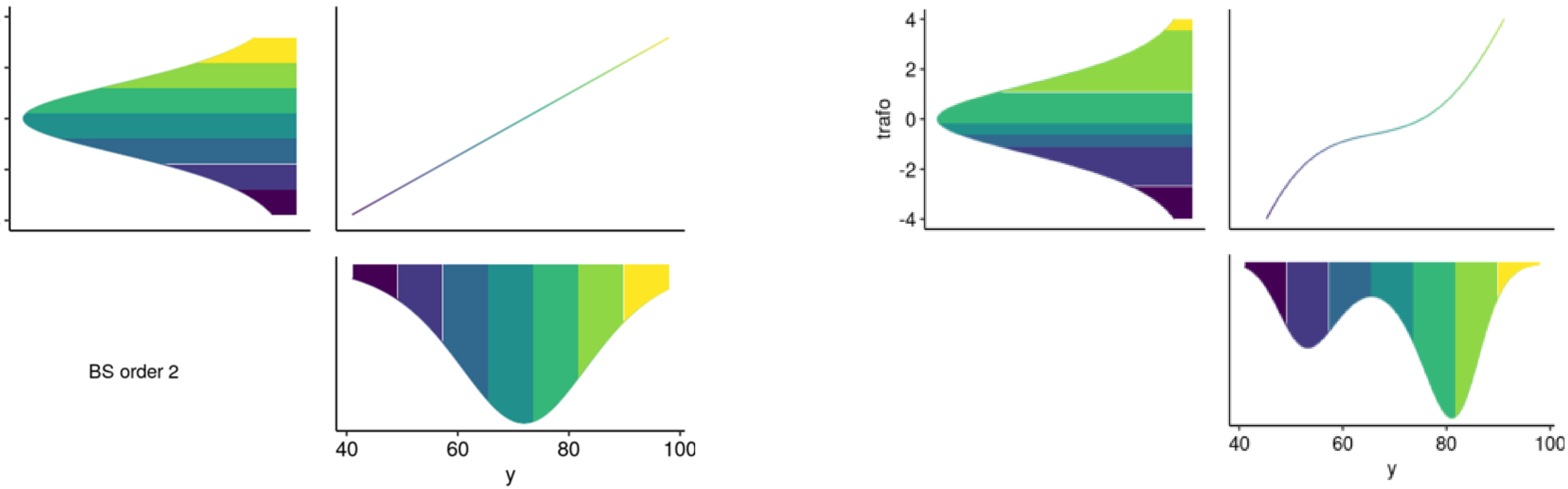
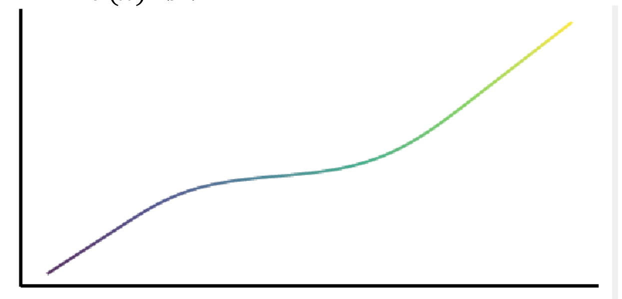


Image credits: Lucas Kook

Using Bernsteinpolynomials to approximate the transformation h

$$z_x = h_{\vartheta(x)}(\tilde{y}) = \sum_{k=1}^{M} \frac{\vartheta_k(x)}{M+1} Be_k(\tilde{y})$$

$\tilde{y} \in [0,1]$

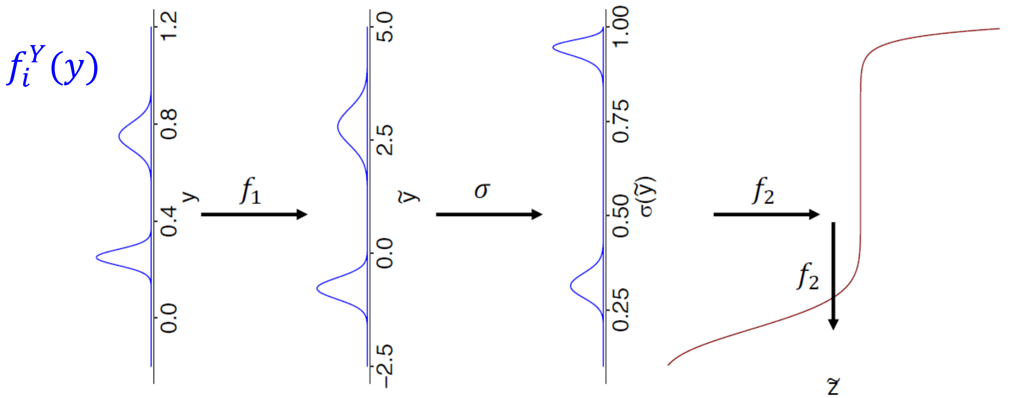


Bernstein polynomials have nice properties:

- They can approximate every function on the support $[0; 1]$
- The order M controls the flexibility
- Its bijective, i.e. monotone increasing, if parameters $\vartheta_1 \leq \vartheta_2 \leq \cdots \leq \vartheta_M$

Most Likely Transfromation (MLT) 2017 by T.Hothorn, L.Möst, P.Bühlmann https://onlinelibrary.wiley.com/doi/full/10.1111/sjos.12291

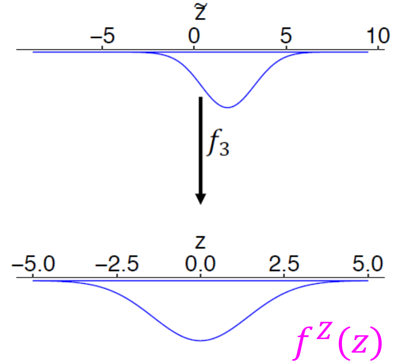
Our deep transformation model



f_1: $\tilde{y} = a(x) \cdot y - b(x)$

f_2: $\tilde{z} = \frac{1}{M+1} \sum_{i=0}^{M} \mathrm{Be}_i(\sigma(\tilde{y}))\vartheta_i(x)$

f_3: $z = \alpha(x) \cdot \tilde{z} - \beta(x)$



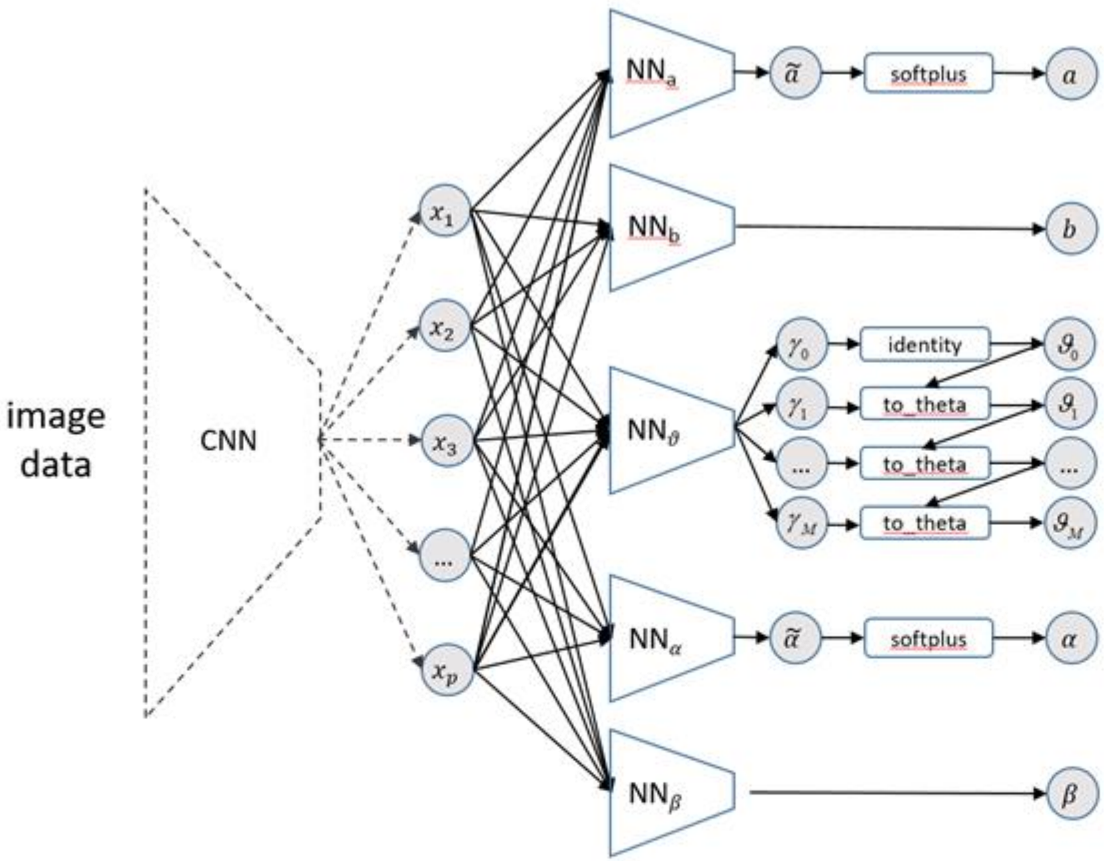
$$h_{\theta(x)} = f_{3,\alpha_x,\beta_x} \circ f_{2,\vartheta_{0_x},\ldots,\vartheta_{M_x}} \circ \sigma \circ f_{1,a_x,b_x}$$

$$\mathrm{NLL} = \sum_i -\log\left(f^Z(z_i) \cdot \left| \frac{\partial h_{\widehat{\theta(x_i)}}}{\partial y} \right| \Big|_{y_i} \right)$$

R implementation: https://github.com/tensorchiefs/dl_mlt
Python implementation: https://github.com/MArpogaus/TensorFlow-Probability-Bernstein-Polynomial-Bijector

Architecture of our Deep transformation model

$$h_{\theta(x)} = f_{3,\alpha_x,\beta_x} \circ \ f_{2,\vartheta_{0_x},\dots,\vartheta_{M_x}} \circ \sigma \circ f_{1,a_x,b_x} \qquad \text{NLL} = \sum_i -\log\left(f^z(z_i) \cdot \left|\frac{\partial h_{\widehat{\theta(x_i)}}}{\partial y}\right|\Big|_{y_i}\right)$$



Ensuring a monotone increasing *h*:

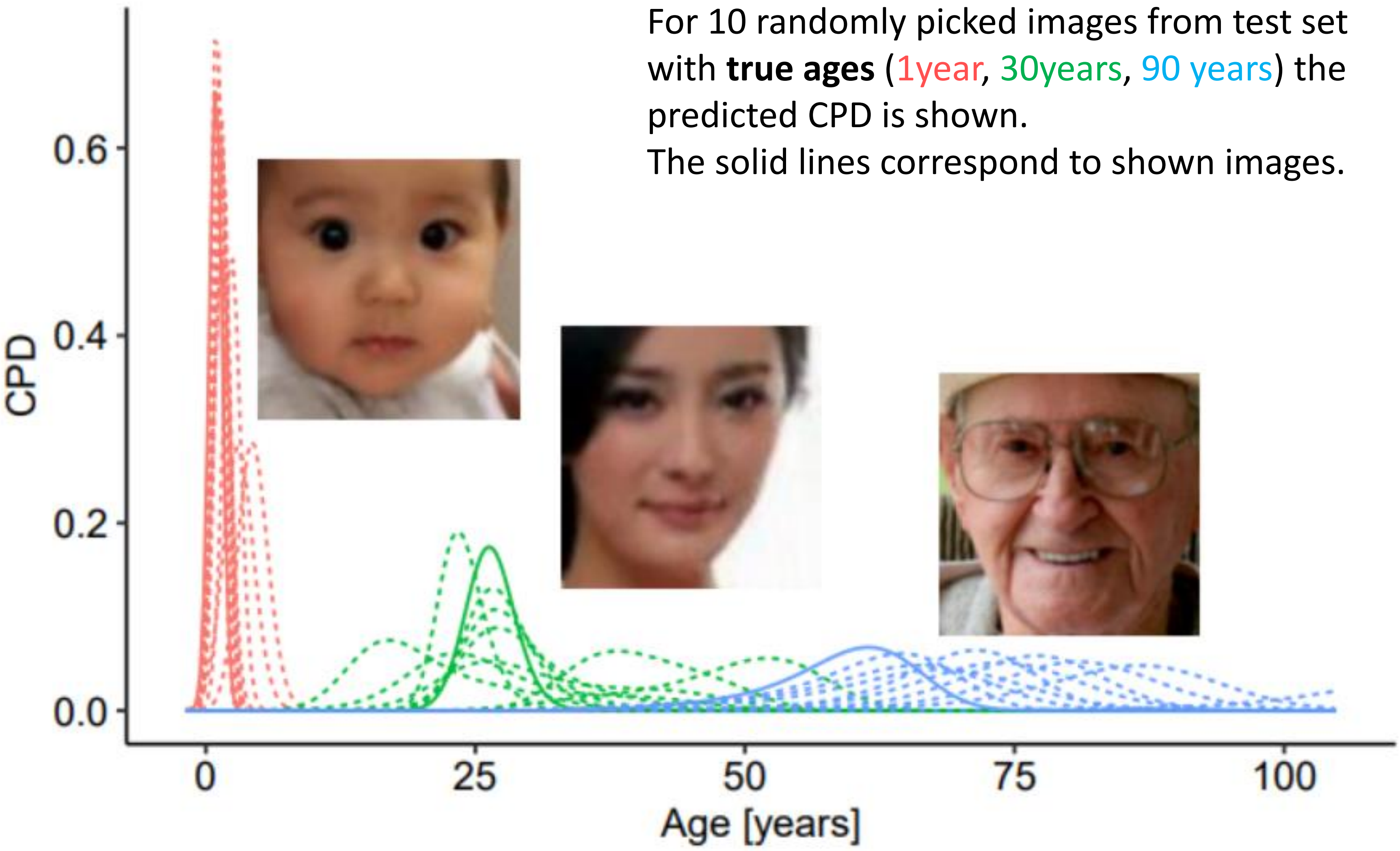
Positive slope:
$a > 0$

Increasing Bernstein coefficients :

$\vartheta_0 = \gamma_0$
$\vartheta_1 = \vartheta_0 + e^{\gamma_1}$
...
$\vartheta_M = \vartheta_{M-1} + e^{\gamma_M}$
$\rightarrow \vartheta_0 < \vartheta_1 \dots < \vartheta_M$

Positive slope:
$\alpha > 0$

Application: Predict CPD for age based on an image

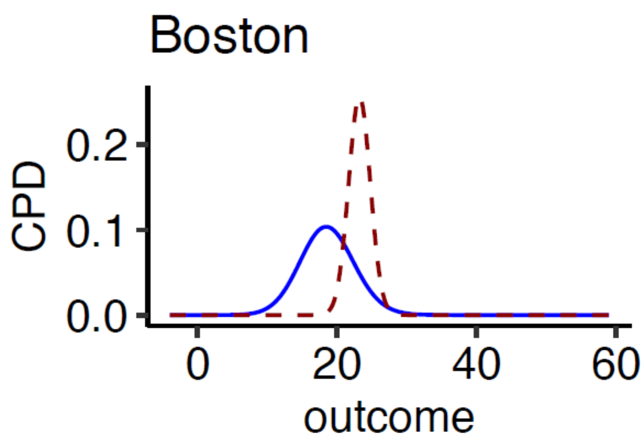


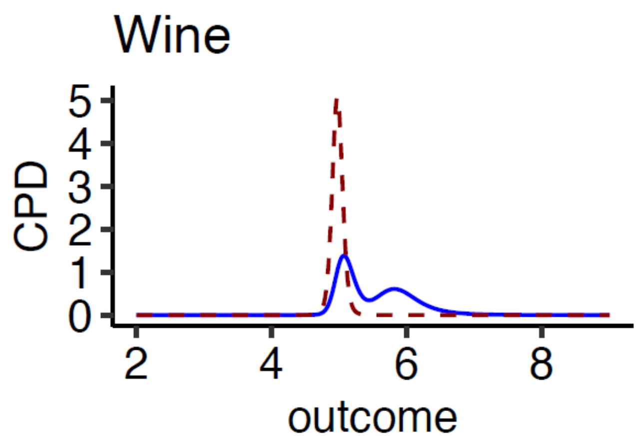
For 10 randomly picked images from test set with **true ages** (1year, 30years, 90 years) the predicted CPD is shown.
The solid lines correspond to shown images.

Application: Benchmarking our model

TABLE I
COMPARISON OF PREDICTION PERFORMANCE (TEST NLL, SMALLER IS BETTER) ON REGRESSION BENCHMARK UCI DATASETS. THE BEST METHOD FOR EACH DATASET IS BOLDED, AS ARE THOSE WITH STANDARD ERRORS THAT OVERLAP WITH THE STANDARD ERRORS OF THE BEST METHOD.

Data Set	N	DL_MLT	NGBoost	MC Dropout	Deep Ensembles	Gaussian Process	MDN	NFN
Boston	506	**2.42 ± 0.050**	**2.43 ± 0.15**	**2.46 ±0.25**	**2.41 ±0.25**	**2.37 ±0.24**	**2.49 ± 0.11**	**2.48 ±0.11**
Concrete	1030	3.29 ± 0.02	**3.04 ± 0.17**	**3.04 ±0.09**	**3.06 ±0.18**	**3.03 ±0.11**	**3.09 ± 0.08**	**3.03 ±0.13**
Energy	768	**1.06 ± 0.09**	**0.60 ± 0.45**	1.99 ±0.09	1.38 ±0.22	**0.66 ±0.17**	**1.04 ± 0.09**	1.21 ±0.08
Kin8nm	8192	-0.99 ± 0.01	-0.49 ± 0.02	-0.95 ±0.03	**-1.20 ±0.02**	-1.11 ±0.03	NA	NA
Naval	11934	**-6.54 ± 0.03**	-5.34± 0.04	-3.80 ±0.05	-5.63 ±0.05	-4.98 ±0.02	NA	NA
Power	9568	**2.85 ± 0.005**	**2.79 ± 0.11**	**2.80 ±0.05**	**2.79 ±0.04**	**2.81 ±0.05**	NA	NA
Protein	45730	**2.63 ± 0.006**	2.81 ± 0.03	2.89 ±0.01	2.83 ±0.02	2.89 ±0.02	NA	NA
Wine	1588	**0.67 ± 0.028**	0.91 ± 0.06	0.93 ±0.06	0.94 ±0.12	0.95 ±0.06	NA	NA
Yacht	308	**0.004 ± 0.046**	**0.20 ± 0.26**	1.55 ±0.12	1.18 ±0.21	0.10 ±0.26	NA	NA





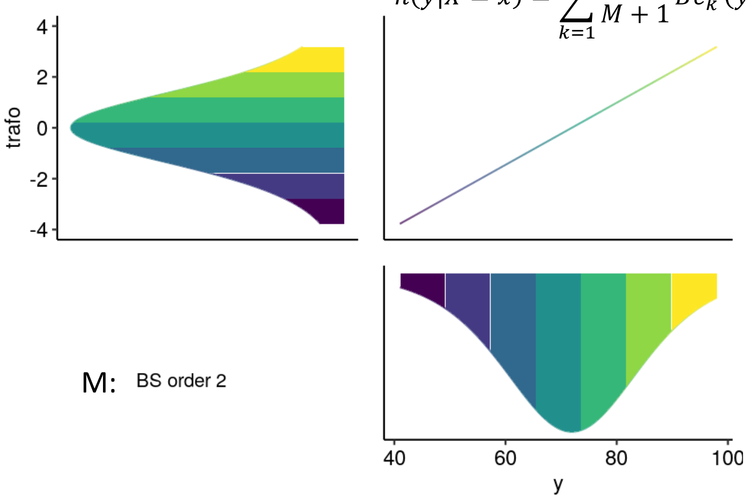
The 2 CPDs (dashed and solid line) correspond to 2 picked observations in the respective data set.

Summary and outlook

- Transformation models allow to model highly flexible outcome
  → extremely high prediction performance
- Any kind of input data and NN architectures can be integrated
- In the mean time we have extended the approach
  - to ordinal outcomes
  - to provide interpretable model parameters

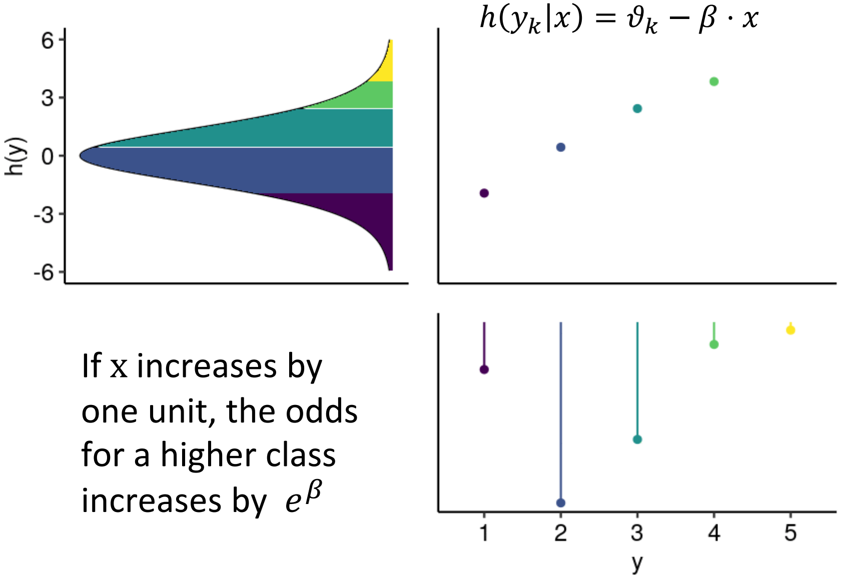
Maximal flexibility but not interpretable:

$$h(\tilde{y}|X=x) = \sum_{k=1}^{M} \frac{\vartheta_k(x)}{M+1} Be_k(\tilde{y})$$



Interpretabel but less flexible:

$$h(y_k|x) = \vartheta_k - \beta \cdot x$$



If x increases by one unit, the odds for a higher class increases by e^{β}

Regression with continuous outcome: https://arxiv.org/abs/2004.00464

Regression with ordinal outcome: https://arxiv.org/abs/2010.08376