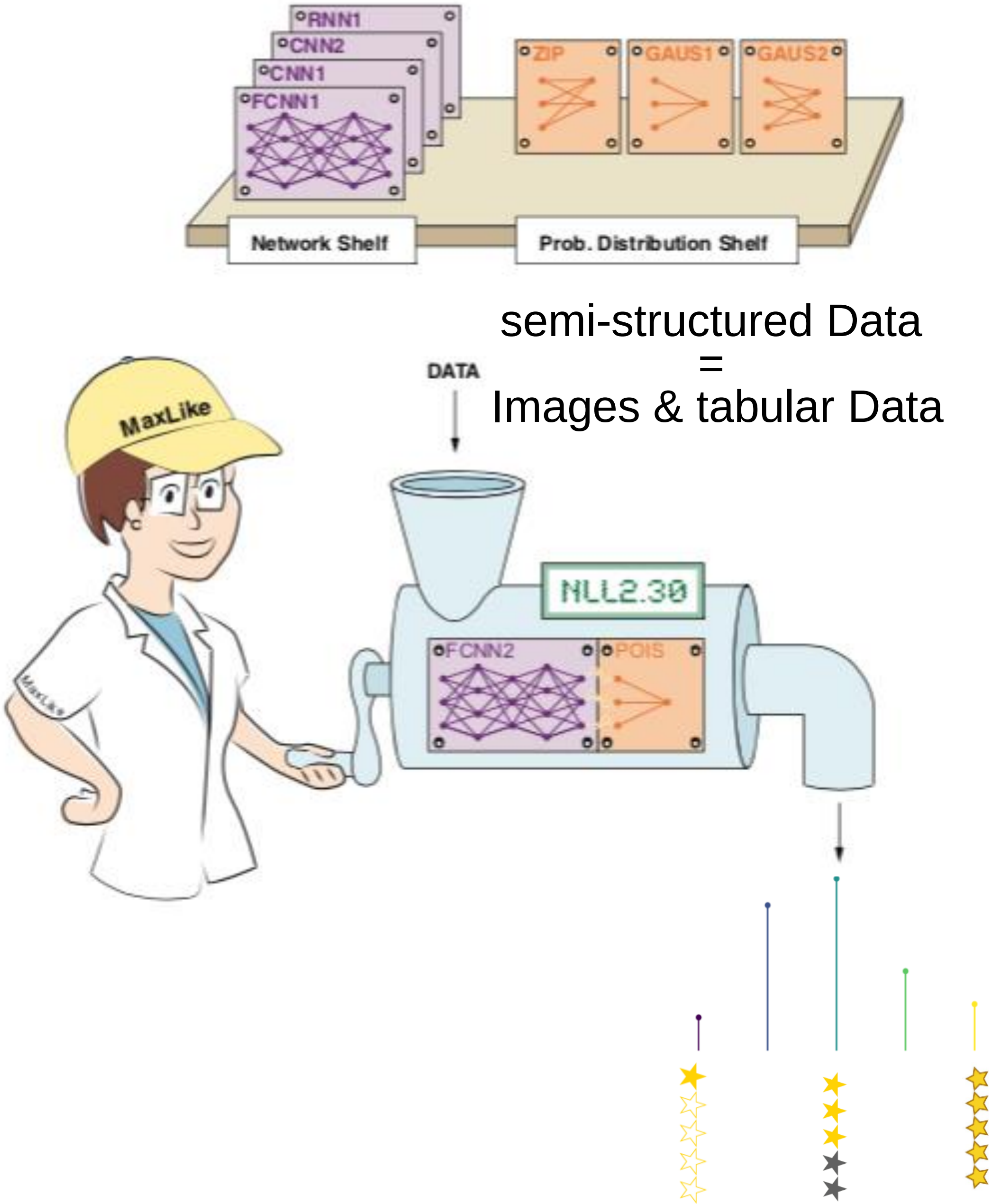
WBL Deep Learning:: Lecture 5

Beate Sick, Oliver Dürr

Deep Learning for interpretable semi-structured models

Zürich, 10/10/2022

Probabilistic DL with semi-structured data



semi-structured Data
=
Images & tabular Data

Aim: Interpretable models for semi-structured data

Use NN for complex data (e.g. images) combined with a statistical model which allows for interpretable coefficients

Today

- Look again at logistic regression as latent variable model
- Logistic regression with semi-structured data
- Interpretable semi-structured Ordinal Regression models (ONTRAMs)
- Interpretable Ordinal Regression Models with Neural Networks
- Formulation as transformation model
- Ensembling for improving prediction performance and quantifying parameter uncertainty

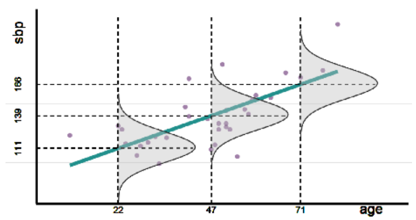
Taking the best of both worlds

Statistics

+ transparent & interpretable

+ valid uncertainty measures

- needs tabular data

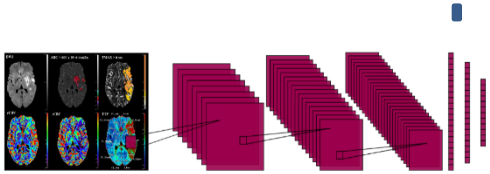


Deep Learning

+ can handle tabular (structured) & unstructured (e.g. image) data

+ high prediction performance

- black box, not interpretable



Interpretable & probabilistic DL

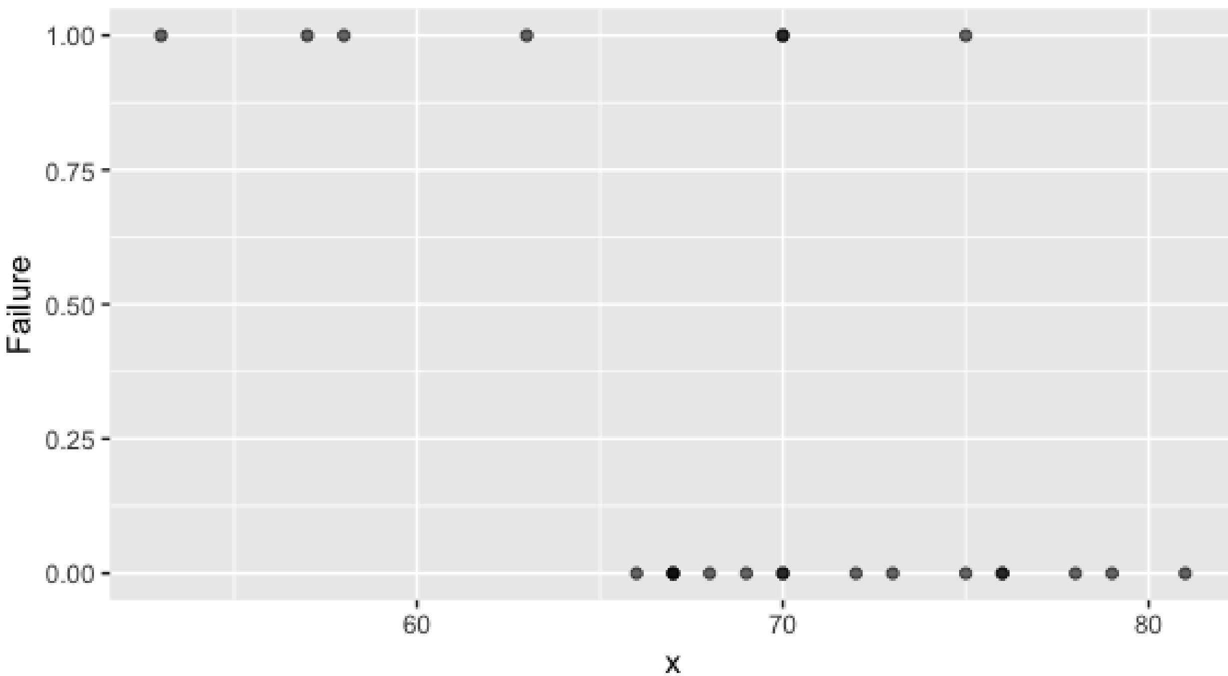
Logistic regression as latent variable model

Recap: Modelling with logistic regression

Zero / one classification

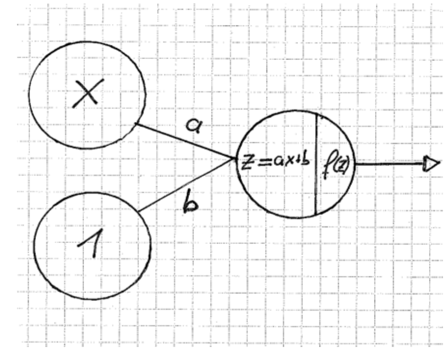
Want: $p(x) = Pr(Y = 1|x)$

Prob. for an O-ring to be defect Y=1 at a given temperature X



- **Guess curve?**

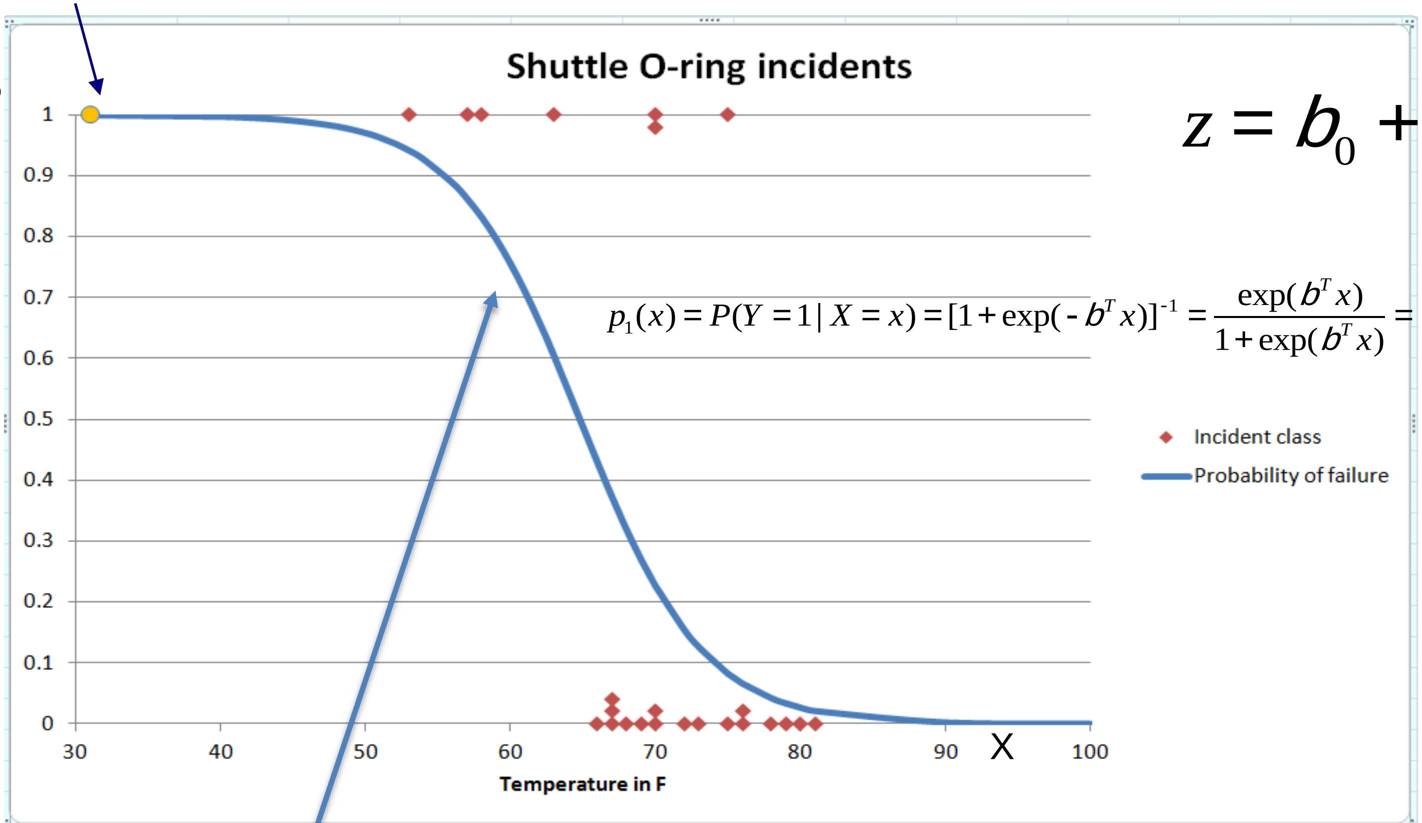
Recap: Logistic Regression



Predict if O-Ring is broken, depending on temperature

Challenger launch @31 F

Prob. of a failure=0.9997



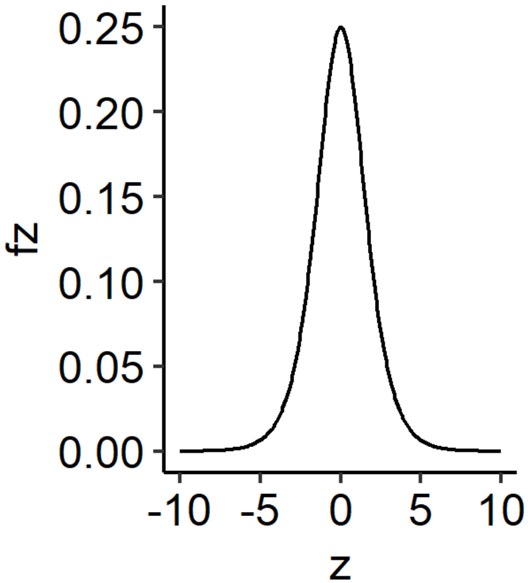
$$z = b_0 + b_1 x$$

$$p_1(x) = P(Y=1 \mid X=x) = [1+\exp(-b^T x)]^{-1} = \frac{\exp(b^T x)}{1+\exp(b^T x)} = f(b^T x)$$

- **What is that curve?**

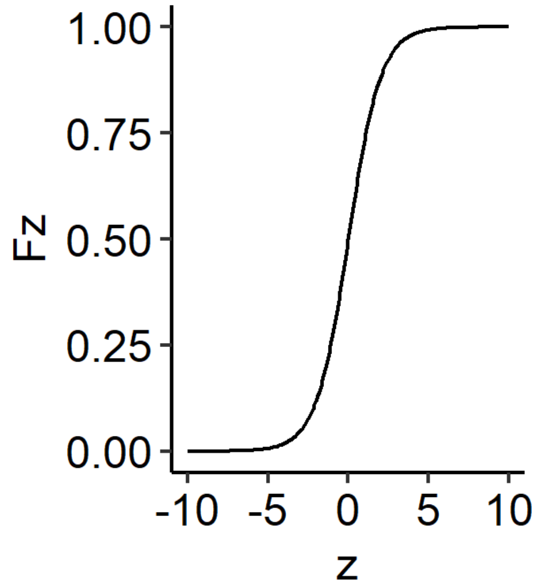
Recap: Standard Logistic Distribution

PDF (f_Z)



$$f_Z(z) = \frac{e^z}{(1+e^z)^2}$$

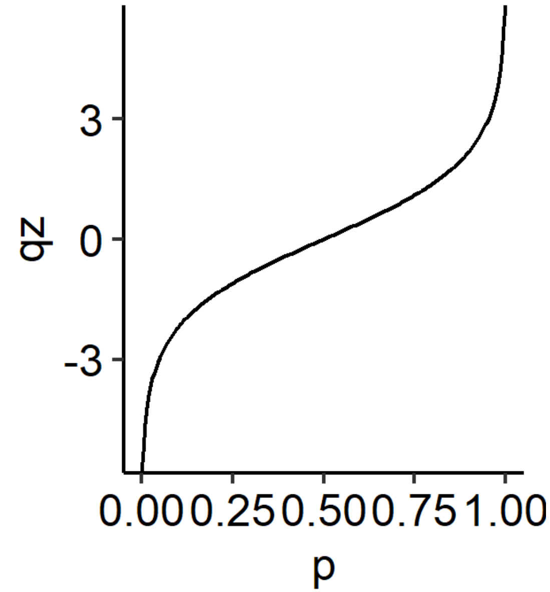
CDF (F_Z)



$$F_Z(z) = \frac{1}{1+e^{-z}}$$

$F_Z(z)$ =expit(z)

Quantiles function (F_Z^{-1})



$$F_Z^{-1}(p) = \log \frac{p}{1-p}$$

$$\begin{aligned} F_Z^{-1}(p) &= \log(\text{odds}) \\ &= \text{logit}(p) \end{aligned}$$

Interpretation with odds

https://www.wincomparator.com/roger-federer-id2930-rafael-nadal-id2905/

- Probability for event
  - $P(Y=1|x) = p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$
  - $odds(x) := \frac{p(x)}{1-p(x)} = e^{\beta_0+\beta_1 x} \in [-\infty, +\infty]$
  - $\log\left(odds(x)\right) = \beta_0 + \beta_1 x$

- Interpretation (odds):
  - Examples
    - Odds : „Prob for winning“ / „Prob for not winning“
      - Federer against Nadal 2:1 Two times more likely that Federe wins
    - Odds : „Probability of broken o-ring“ vs. Non-broken
      - 8:1

Odds ratios

- $\log\left(odds(x)\right) = \beta_0 + \beta_1 x$

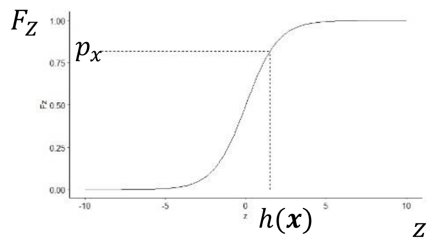
- Q: How does the odds change with x
- Example Tennis (Federer against Nadal)
  - $x = 0$ (not grass) $odds(x = 0) = \exp(\beta_0) = 1.2$
  - $x = 1$ (grass) $odds(x = 1) = \exp(\beta_0 + \beta_1 \cdot 1) = 2$
  - Odds Ratio: $OR_{x=0 \to x=1} = \frac{odds(x=1)}{odds(x=0)} = \exp(\beta_1 \cdot 1) = 2/1.2$
  - $\log(OR_{x=0 \to x=1}) = \beta_1 = 2/1.2$

- Works also for continuous x important is just Δx (not absolute value)

The logistic regression model

Model for the cut-point:

$$h(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \;\rightarrow$$



$$p_{x_i} = F_Z(h(\boldsymbol{x}_i))$$
$$\Updownarrow$$
$$h(\boldsymbol{x}_i)) = F_Z^{-1}(p_{x_i})$$

Using the logistic distribution: $F_Z(z) = F_L(z) = \dfrac{1}{1+e^{-z}} \;\Leftrightarrow\; F_Z^{-1}(p) = \log\dfrac{p}{1-p}$

$$h(\boldsymbol{x}) = F_L^{-1}(p_{\boldsymbol{x}}) = \log\left(\frac{p_{\boldsymbol{x}}}{1-p_{\boldsymbol{x}}}\right) = \log(\text{odds}(\boldsymbol{x})) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

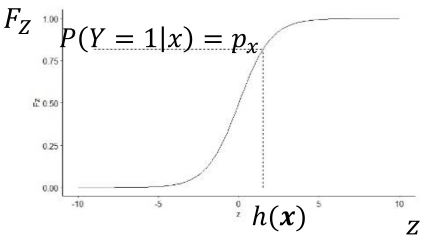
$$\text{odds}(\boldsymbol{x}_i) = e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p} = e^{\beta_0} \cdot e^{\beta_1 x_1} \cdot \ldots \cdot e^{\beta_k (x_k+1)} \ldots \cdot e^{\beta_p x_p}$$

$$p_{x_i} = F_L(h(\boldsymbol{x}_i)) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p)}}$$

Modelparameters are interpretable as log-odds ratio

Model for cut-point:

$$\log(\text{odds}(\boldsymbol{x})) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$



The **coefficient** β_k as the **log-odds-ratio** for $Y = 1$ when comparing a situation where x_k is increases by 1 unit (while fixing all other variables) with the situation before increasing x_k

$$\log(\text{OR}_k) = \log\left(\frac{\text{odds}(x_1,\ldots,x_k+1,\ldots,x_p)}{\text{odds}(x_1,\ldots,x_k,\ldots,x_p}\right) = \log\left(\frac{e^{\beta_0} \cdot e^{\beta_1 x_1} \cdot \ldots \cdot e^{\beta_k (x_k+1)} \ldots \cdot e^{\beta_p x_p}}{e^{\beta_0} \cdot e^{\beta_1 x_1} \cdot \ldots \cdot e^{\beta_k x_k} \ldots \cdot e^{\beta_p x_p}}\right) = \log\left(e^{\beta_k}\right) = \beta_k$$

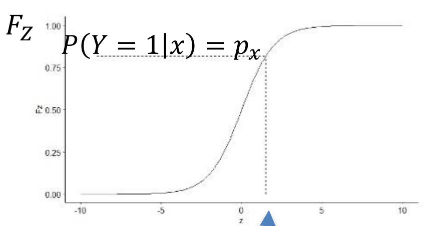
$$\Rightarrow \quad e^{\beta_k} = \text{OR}_{x_k \to x_k + 1}$$

Logistic Regression as latent variable model

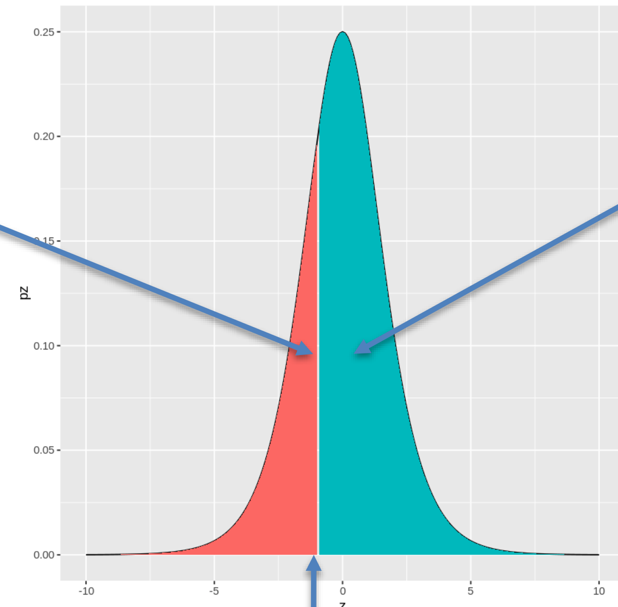
Idea:
A continuous latent (unobserved) variable *z* determines the probability to observe $Y = 1$

P(Y=1)

P(Y=0)



$z = h(x) = \beta_0 + \beta_1 \cdot x$



$z = h(x) = \beta_0 + \beta_1 \cdot x$ is cut-point

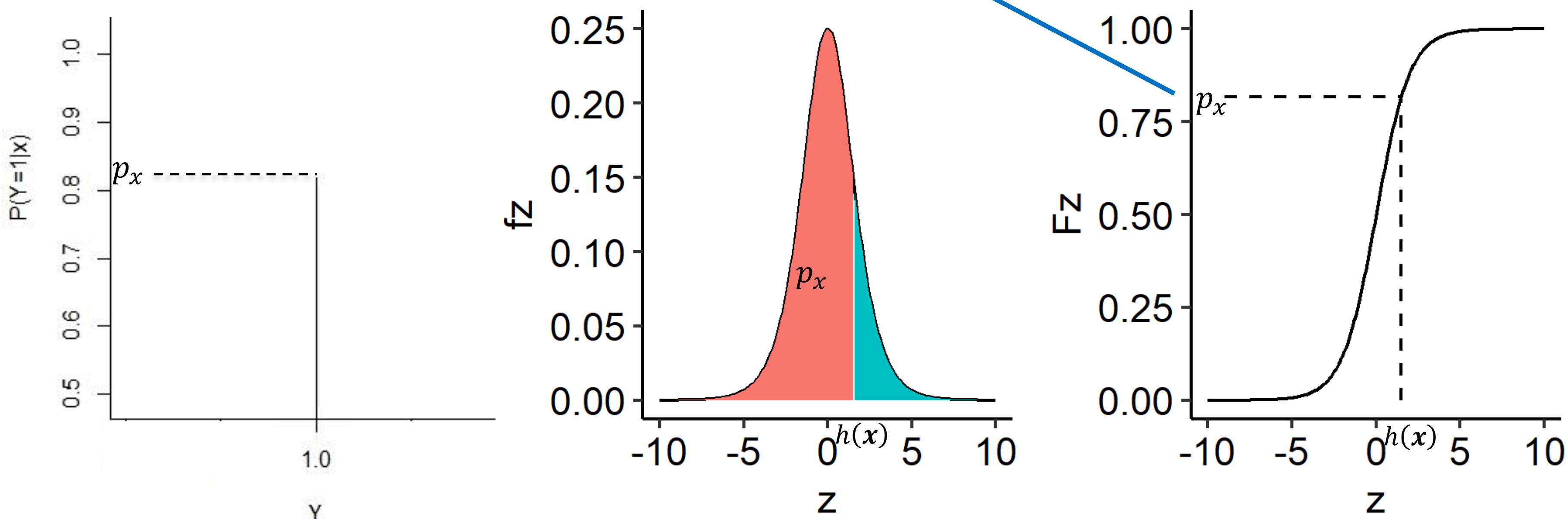
Logistic Regression as latent variable model

We fit a logistic regression model $(Y|x) \sim \text{Ber}(p_{\boldsymbol{x}})$ by minimizing the NLL

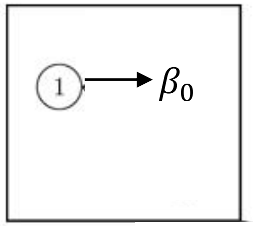
$$\text{NLL} = -\frac{1}{n}\sum_i \log(L_i), \quad \log(L_i) = \log\big(P(Y = y_i|\boldsymbol{x}_i)\big) = y_i \cdot \log\big(p_{\boldsymbol{x}_i}\big) + (1 - y_i) \cdot \log\big(1 - p_{\boldsymbol{x}_i}\big)$$

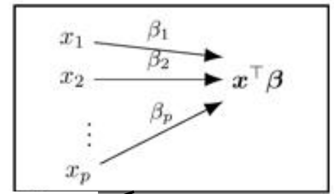
In DL we minimize the NLL by finding the optimal β via SGD (stochastic gradient decent)

$$h(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \quad \rightarrow \quad \boldsymbol{p_{x_i} = P(Y = 1|x) = F_Z(h(x_i))}$$

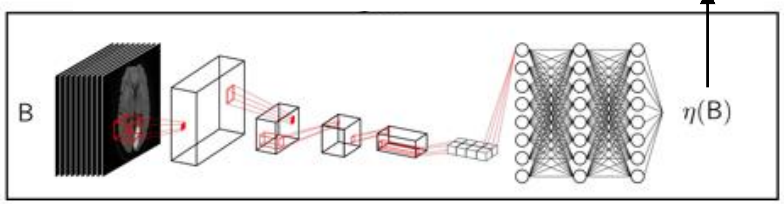


Semi-structured logistic regression via DL approach





$$h(x, B) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \eta(B)$$



Jointly train all NNs by minimizing the NLL loss:

$$\mathrm{NLL} = -\frac{1}{n}\sum_i \log(L_i), \qquad L_i = F_L\big(h(x,B)\big) = \frac{1}{1+e^{-h(x,B)}}$$

Parameters are still interpretable as log-odds ratios: $\hat{\beta}_i = \log(\mathrm{OR}_{x_i \to (x_i+1)})$

Kook & Herzog, Dürr, Hothorn, Sick Pattern Recognition (2021) & https://arxiv.org/abs/2010.08376

Ordinal regression
(more then two levels)

Ordinal Data (Examples)

- Amazon Ratings 
- Medical ratings
  - Reopatiy Score
  - Neuroscore
- Wine quality (score between 0 and 10)
  - UCI-Data* set (N=4898) from (Cortez et. al. 2009) with 11 covariates like:
    - fixed acidity, residual sugar, sulphates, ..., alcohol
- Age group (UKTFace)
  - Images as high dimensional features

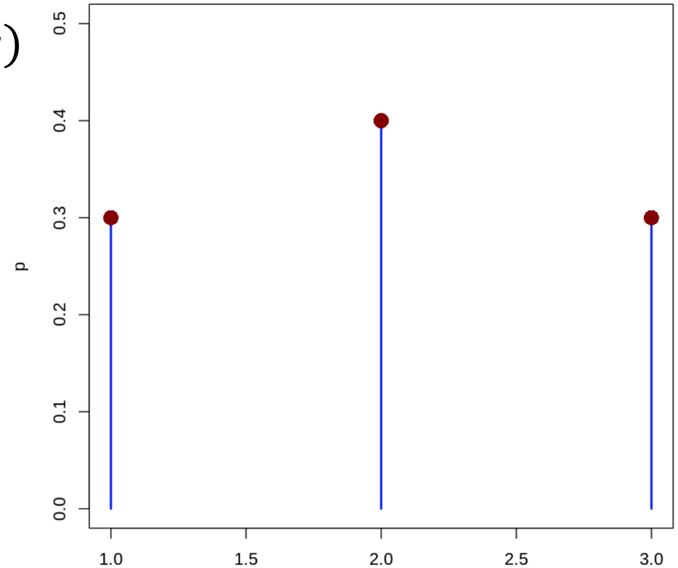
«Less than numerical data, more than just classes / categories»

Goal of ordinal regression

- Get the conditional probability distribution CPD for a given x.
  - $p(y|x)$

- Simple example
  - x alcohol content in wine
  - y=1,2,3   corresponds to bad, medium, good

$p(y|x)$

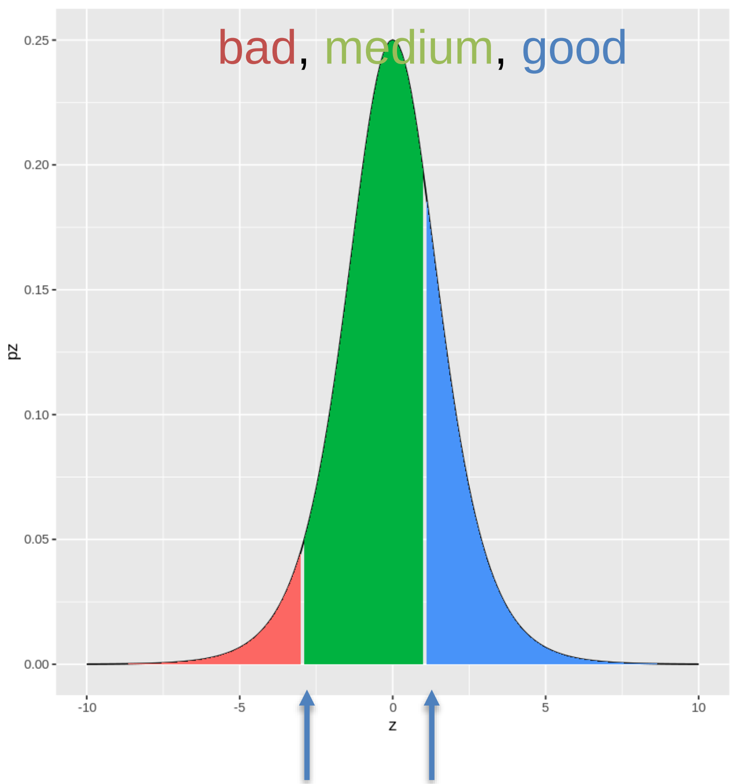


y=1 y=2 y=3

Distribution for fixed x

Proportional odds-model

Latent variable model can be extended for more than 2 levels



$h_1 = \vartheta_1 - x \cdot \beta$ $\quad$ $h_2 = \vartheta_2 - x \cdot \beta$

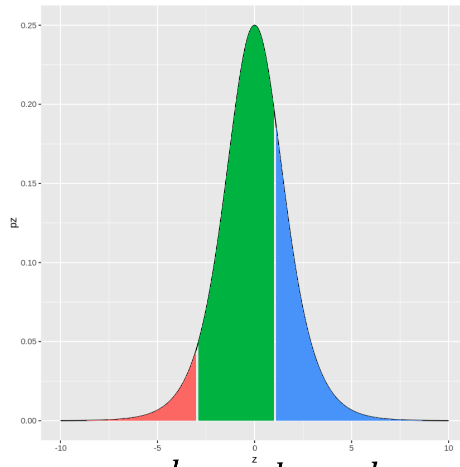
If x changes, all cut points are moving together.

Cut-points modeled via: $h_i = \vartheta_i - x \cdot \beta$

*Gerhard Tutz. Regression for categorical data, volume 34. Cambridge University Press, 2011

Proportional odds model (interpretation)

bad, medium, good

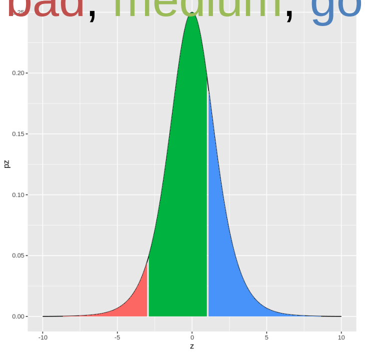


h_1 h_2 $h_3 = \infty$

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

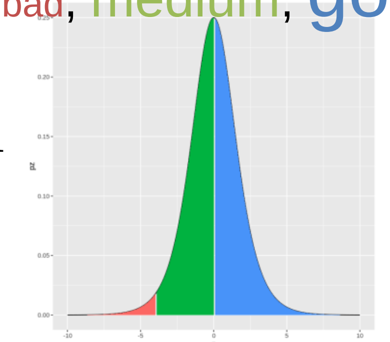
- Since we have more levels, odds are now defined as
  - $odds(Y > y_k|x) = \frac{P(Y>y_k|x)}{P(Y\leq y_k|x)}$
  - Example $k = 2$ odds prob for good / prob less than good
- Changes in odds with x again with odds ratio
  - $OR_{x \to x+1} = \frac{odds(Y>y_k|x+1)}{odds(Y>y_k|x)} = e^{\beta}$ (not depending on k)
- Example $x = alcohol$

bad, medium, good



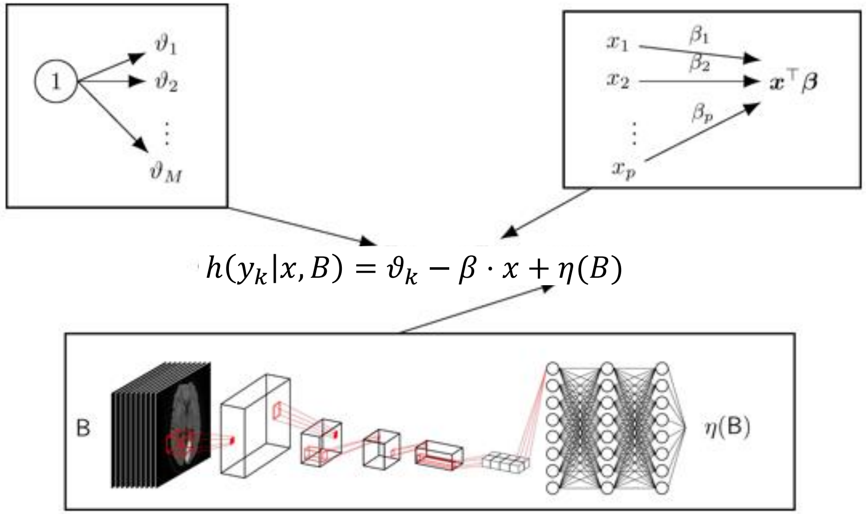
$x \to x + 1$

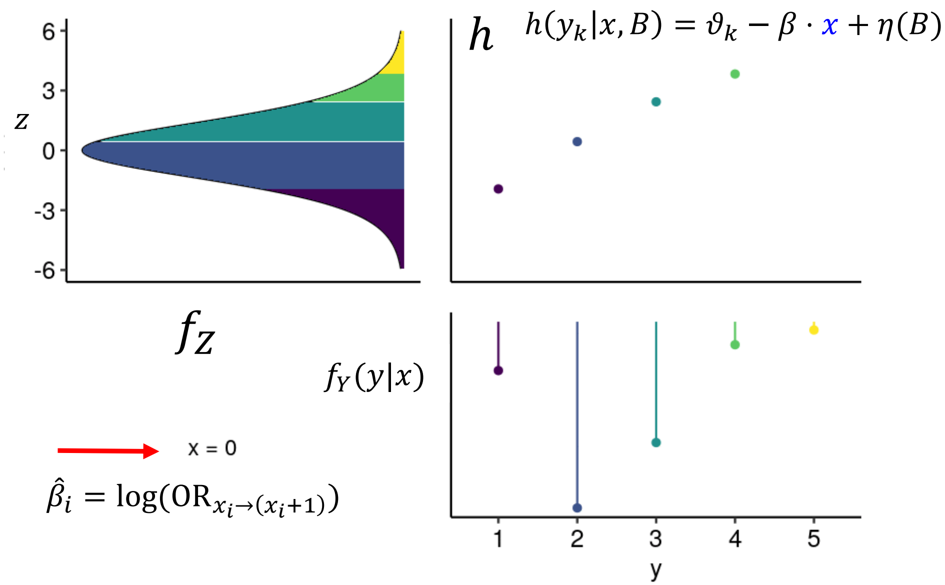
bad, medium, good



- Questions: Does alcohol make the wine taste better?
- What is β in $h_i = \vartheta_i - x \cdot \beta$

Semi-structured ordinal regression (ONTRAMs) via DL





Jointly train NNs with NLL loss:

$$\mathrm{NLL} = -\frac{1}{n}\sum_i \log(L_i)$$

$$\begin{aligned}\mathcal{L}_i(h; y_{ki}, \boldsymbol{x}_i) &= \mathbb{P}(Y = y_{ki}|\boldsymbol{x}_i) = F_Y(y_{ki}|\boldsymbol{x}_i) - F_Y(y_{(k-1)i}|\boldsymbol{x}_i) \\ &= F_Z(h(y_{ki}|\boldsymbol{x}_i)) - F_Z(h(y_{(k-1)i}|\boldsymbol{x}_i)).\end{aligned}$$

Kook & Herzog, Dürr, Hothorn, Sick Pattern Recognition (2021) & https://arxiv.org/abs/2010.08376

UKT-Face

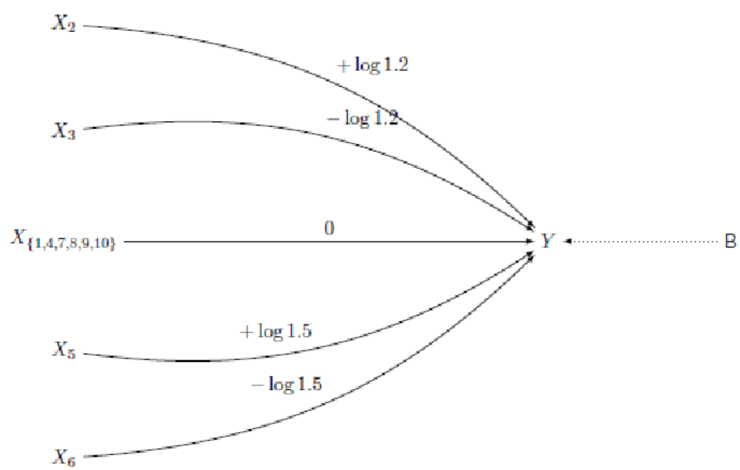
Ordinal neural network regression to model ordinal outcome is the **age-category**.

Modeled transformation function: $h(y_k|x, \mathrm{B}) = \vartheta_k - x^\top \beta - \eta(\mathrm{B})$

Image input Data (one random example for each of the 7 age-categories):

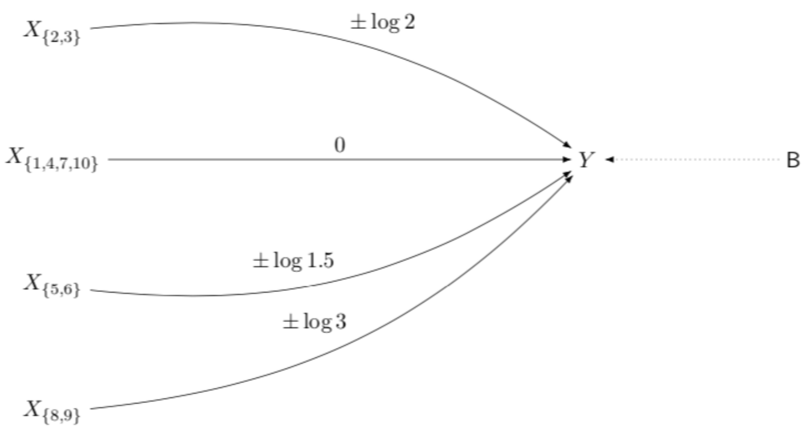
Tabular Co-Variates:

- Gender
- Ethnicity
- 10 simulated covariates with known effect sizes

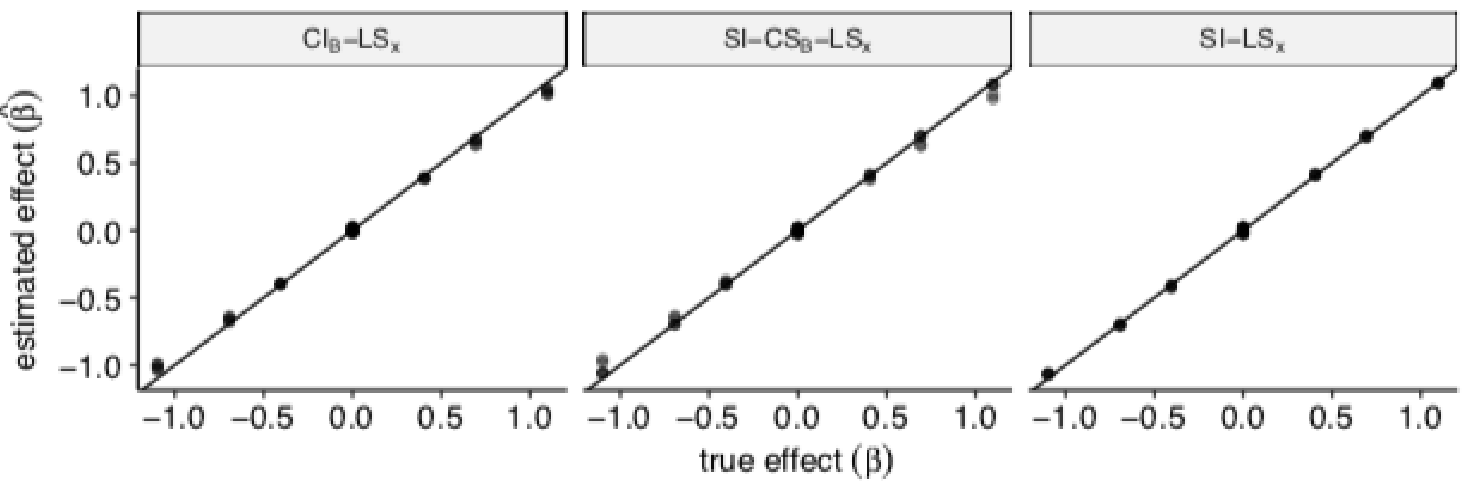


https://colab.research.google.com/drive/17p3I2p_PctFptAlQNZOfrHx1YPhyQ5yn#scrollTo=A7lojWmkbmBe

UKT-Face: Can, we recover simulated tabular data?



C



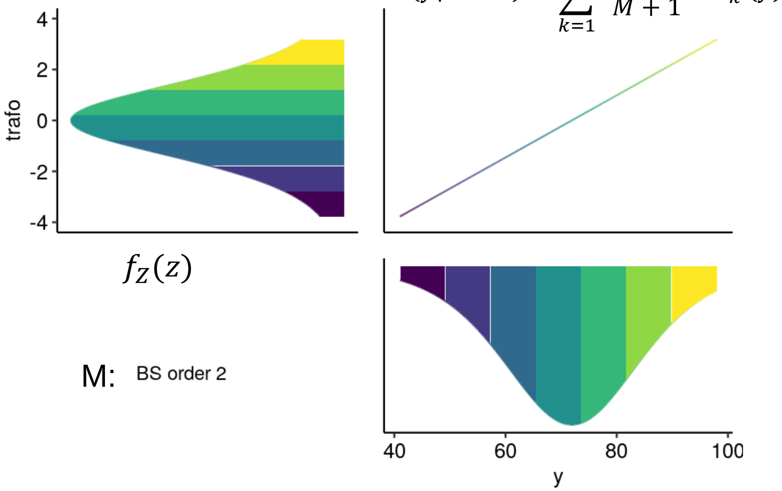
Recovering the simulated effects.
Practical example. Ongoing…

Transformation models for flexible distributional regression

- In traditional regression, the family of the conditional outcome distribution $F_Y(y|x, B \ldots)$ is predefined, and the parameters of this distribution are fitted.
- In TMs the conditional outcome distribution $F_Y(y|x, B \ldots)$ is achieved by transforming a parameter-free distribution $F_Z(z = h(y|x, B \ldots))$ requiring to estimate the parameters of the conditional transformation function $h(y|x, B \ldots)$.

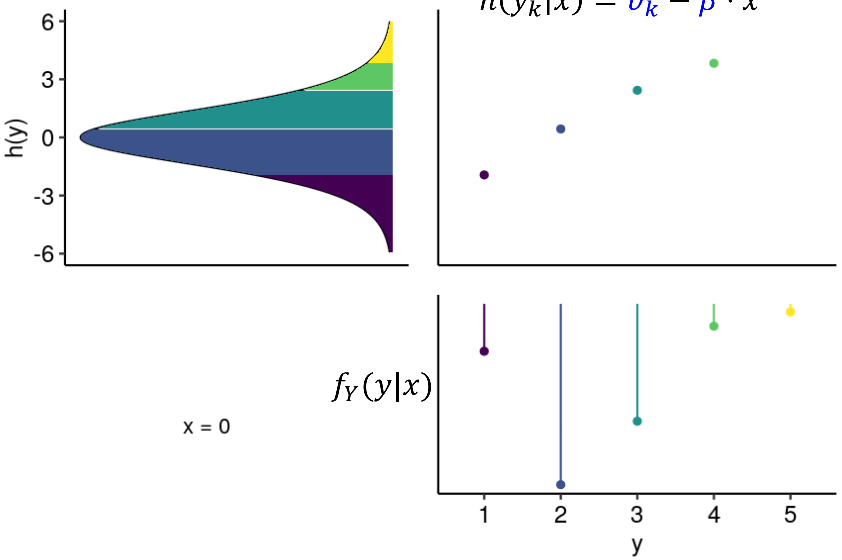
Continuous outcome → continuous *h*

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



Discrete outcome → discrete *h*

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



Sick, Hothorn, Dürr (2020): https://arxiv.org/abs/2004.00464

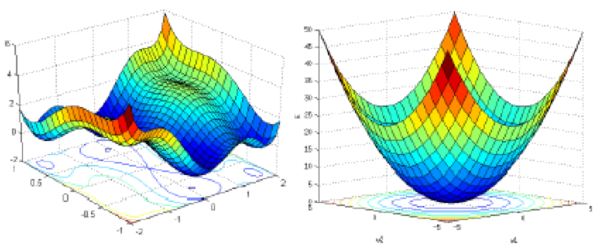
Kook & Herzog, Dürr, Hothorn, Sick (2021) https://arxiv.org/abs/2010.08376

Ensembling

Classical "Deep Ensembles" as used in deep learning

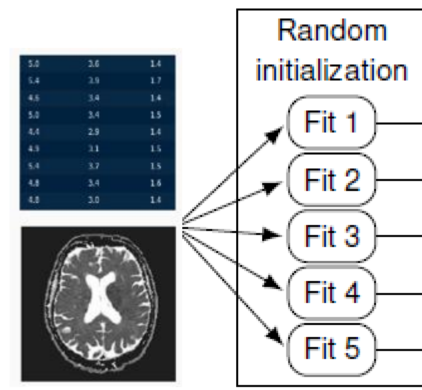
Refitting a deep NN with same data but new random initialization yields slightly different parameter estimates.

Optimization is non-convex:



Reasons:
– Over-parametrization
– Training on mini-batches
– Random weight initialization

Constructing deep ensembles:



Predicted class probabilities

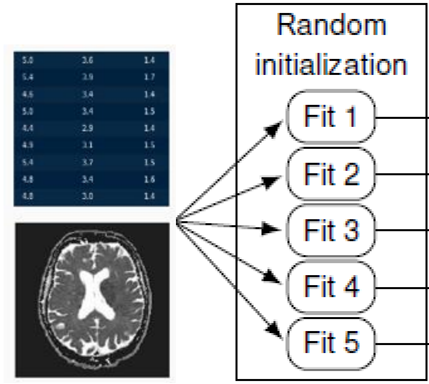
C1	C2	C3	C4	C5	C6
0.1	0.1	0.2	0.4	0.1	0.1
0.05	0.05	0.2	0.3	0.2	0.2
0.2	0.2	0.1	0.05	0.05	0.4
0.05	0.05	0.05	0.05	0.2	0.6
0.3	0.25	0.25	0.1	0.1	0

- Column mean:
  point estimate for class probability

- Column standard deviation:
  uncertainty of point estimate

Deep Ensembles have a higher prediction performances than an average single NN.

Interpretable transformation Ensembles



Interpretable parameters in h

θ_1	θ_2	θ_3	β_1	β_2	η
1.2	1.8	2.3	-1.8	2.5	4.5
1.1	1.7	2.1	-1.6	2.5	4.7
1.3	1.8	2.2	-1.5	2.7	4.2
1.2	1.5	1.9	-1.9	2.4	4.3
1.3	1.6	2.3	-1.7	2.6	4.4

- Column mean:
  point estimate for parameters
- Column standard deviation:
  uncertainty for parameters

Transformation Ensembles yield interpretable parameters along with uncertainty estimates and have higher performance than single deep transformation models.

$$\bar{F}^{\mathrm{t}}_{\mathrm{M}} = F_Z\left(\frac{1}{M}\sum\nolimits_{m=1\ldots M} h_m(y|x,B)\right) = F_Z\left(\bar{\theta}_k - x^\top \bar{\beta} - \bar{\eta}(B)\right)$$

Kook, Götschi , Baumann3 , Hothorn, Sick (2022) https://arxiv.org/abs/2205.12729

Summary on ordinal neural transformation networks (ONTRAMs)

- ONTRAM allows to work with image data and tabular predictors

- ONTRAM allows for the same interpretability than statistical ordinal regression
  - F_Z=logistic and $h(y_k|x) = \vartheta_k + \sum_{l=1}^{p} \beta_l \cdot x_l$ with log-odds interpretation of β

- ONTRAM has the high prediction performance of DL models

- Via transformation ensembles, the prediction performance can be further improved and the uncertainty of the interpretable parameters are quantified.