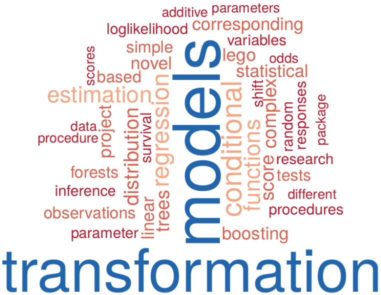
Ensembling deep transformation models

Advances in statistical modeling with neural networks

Lucas Kook, Andrea Goetschi, Philipp FM Baumann,
Torsten Hothorn, Beate Sick



CMStatistics 2022
December 16, 2022

Kook_Lucas
LucasKook

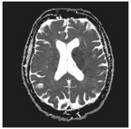
Setup

Available data:

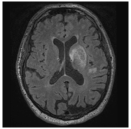
Tabular

5.0	3.6	1.4
5.4	3.9	1.7
4.6	3.4	1.4
5.0	3.4	1.5
4.4	2.9	1.4
4.9	3.1	1.5
5.4	3.7	1.5
4.8	3.4	1.6
4.8	3.0	1.4

Image



Image

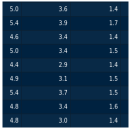


Text

Setup

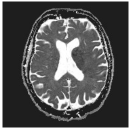
Available data:

Tabular

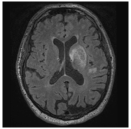


5.0	3.6	1.4
5.4	3.9	1.7
4.6	3.4	1.4
5.0	3.4	1.5
4.4	2.9	1.4
4.9	3.1	1.5
5.4	3.7	1.5
4.8	3.4	1.6
4.8	3.0	1.4

Image



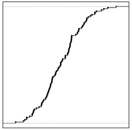
Image



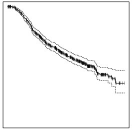
Text

Response:

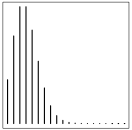
Continuous



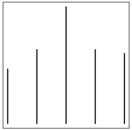
Survival



Count



Ordinal

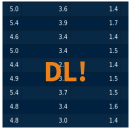


How do changes in the predictors propagate to the **distribution** of the response?

Setup

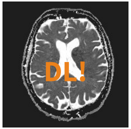
Available data:

Tabular

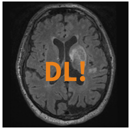


5.0	3.6	1.4
5.4	3.9	1.7
4.6	3.4	1.4
5.0	3.4	1.5
4.4	2.[illegible]	1.4
4.9	3.[illegible]	1.5
5.4	3.7	1.5
4.8	3.4	1.6
4.8	3.0	1.4

Image



Image

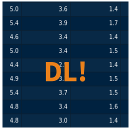


Text

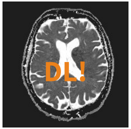
Setup

Available data:

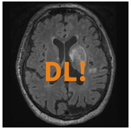
Tabular



Image



Image



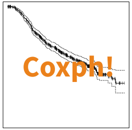
Text

$\longrightarrow$

Response:

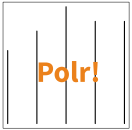
Continuous

Survival



Count

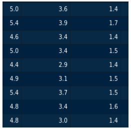
Ordinal



Setup

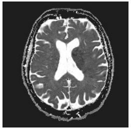
Available data:

Tabular

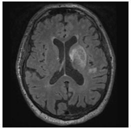


5.0	3.6	1.4
5.4	3.9	1.7
4.6	3.4	1.4
5.0	3.4	1.5
4.4	2.9	1.4
4.9	3.1	1.5
5.4	3.7	1.5
4.8	3.4	1.6
4.8	3.0	1.4

Image



Image

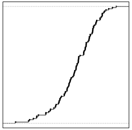


Text

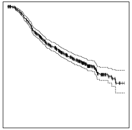
How can we handle non-tabular data?

Response:

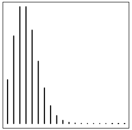
Continuous



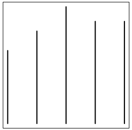
Survival



Count



Ordinal



How can we cover all response types?

A bird's eye view on regression

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(\cdot) := \mathbb{P}(Y \leq \cdot \mid \boldsymbol{X} = \boldsymbol{x})$$

- Normal linear regression

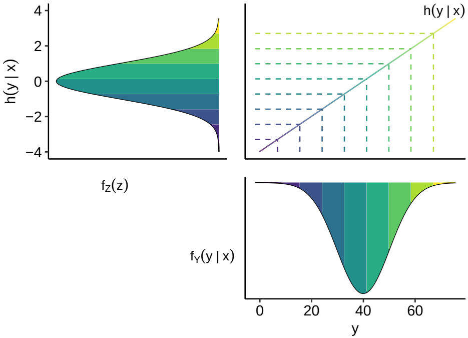
$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y) = \Phi(\sigma^{-1}(y - \alpha + \boldsymbol{x}^\top \beta))$$

- Proportional odds logistic regression

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y_k) = \operatorname{expit}(\vartheta_k + \boldsymbol{x}^\top \beta)$$

- Cox proportional hazards model

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y) = 1 - \exp(-\exp(\log \Lambda_0(y) + \boldsymbol{x}^\top \beta))$$



A bird's eye view on regression

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(\cdot) := \mathbb{P}(Y \leq \cdot \mid \boldsymbol{X} = \boldsymbol{x})$$

- Normal linear regression

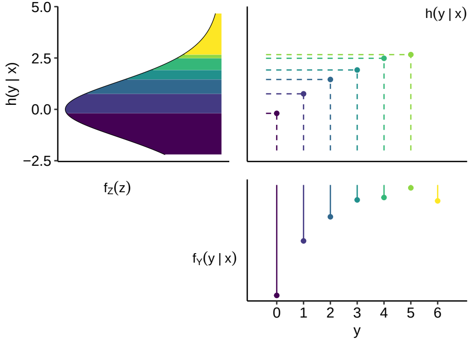
$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y) = \Phi(\sigma^{-1}(y - \alpha + \boldsymbol{x}^\top \beta))$$

- Proportional odds logistic regression

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y_k) = \operatorname{expit}(\vartheta_k + \boldsymbol{x}^\top \beta)$$

- Cox proportional hazards model

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y) = 1 - \exp(-\exp(\log \Lambda_0(y) + \boldsymbol{x}^\top \beta))$$



A bird's eye view on regression

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(\cdot) := \mathbb{P}(Y \leq \cdot \mid \boldsymbol{X} = \boldsymbol{x})$$

- Normal linear regression

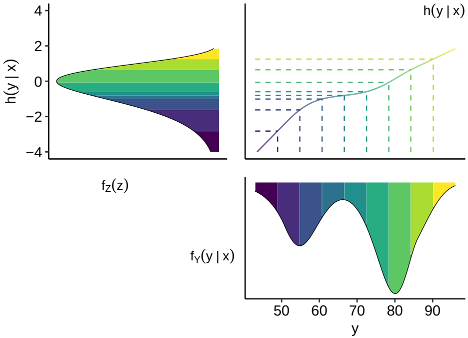
$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y) = \Phi(\sigma^{-1}(y - \alpha + \boldsymbol{x}^\top \beta))$$

- Proportional odds logistic regression

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y_k) = \operatorname{expit}(\vartheta_k + \boldsymbol{x}^\top \beta)$$

- Cox proportional hazards model

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y) = 1 - \exp(-\exp(\log \Lambda_0(y) + \boldsymbol{x}^\top \beta))$$



A bird's eye view on regression

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(\cdot) := \mathbb{P}(Y \leq \cdot \mid \boldsymbol{X} = \boldsymbol{x})$$

- Normal linear regression

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y) = \Phi(\sigma^{-1}(y - \alpha + \boldsymbol{x}^\top \boldsymbol{\beta}))$$

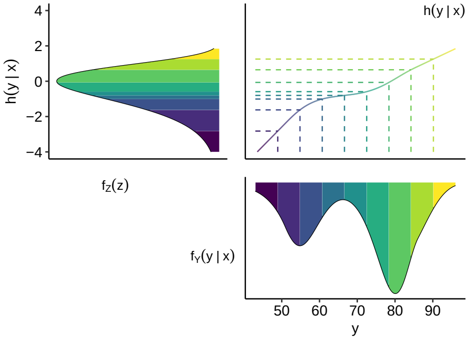
- Proportional odds logistic regression

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y_k) = \operatorname{expit}(\vartheta_k + \boldsymbol{x}^\top \boldsymbol{\beta})$$

- Cox proportional hazards model

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y) = 1 - \exp(-\exp(\log \Lambda_0(y) + \boldsymbol{x}^\top \boldsymbol{\beta}))$$

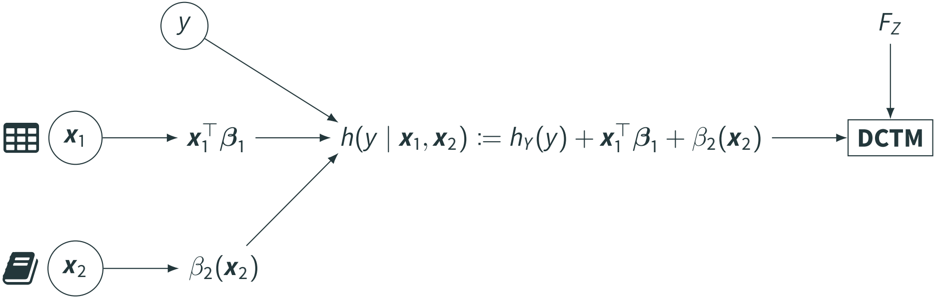
- And many more ...



Transformation models

$$F_{Y|\boldsymbol{X}=\boldsymbol{x}}(y) = F_Z(h_Y(y) + \boldsymbol{x}^\top \boldsymbol{\beta})$$

Example: Parametrization of an additive transformation function

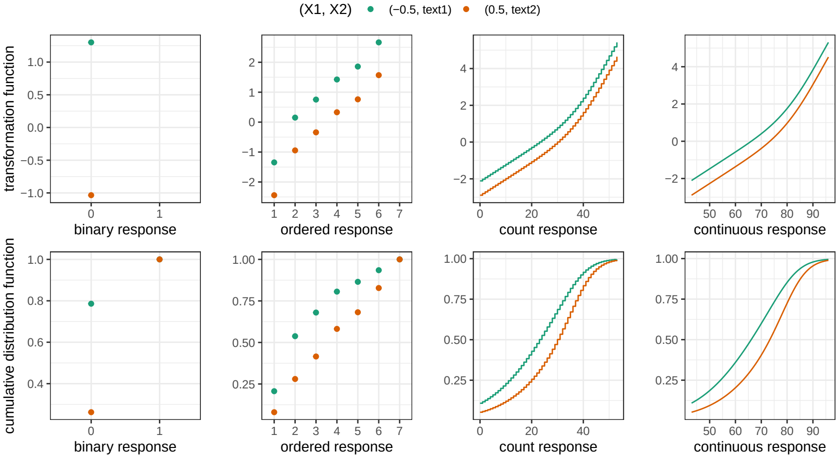


DCTM: Deep conditional transformation model : Tabular data : Text data

10.48550/arXiv.2211.13665

Example: Parametrization of an additive transformation function

$$h(y \mid \boldsymbol{x}_1, \boldsymbol{x}_2) := h_Y(y) + \boldsymbol{x}_1^\top \beta_1 + \beta_2(\boldsymbol{x}_2)$$



10.48550/arXiv.2211.13665

DCTMs are not enough

Individual deep learning models may make **unreliable predictions**

Common criticism:

- No uncertainty quantification
- Small sample sizes
- Stochastic fitting procedure

DCTMs are not enough

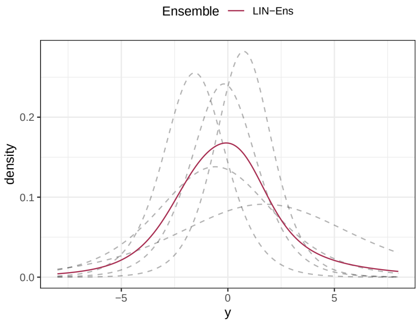
Individual deep learning models may make **unreliable predictions**

Common criticism:

- No uncertainty quantification
- Small sample sizes
- Stochastic fitting procedure

Deep ensembles improve prediction

1. Fit M instances of the same model
2. Average their M predictions



$$\bar{F}^{M}_{Y|\boldsymbol{X}=\boldsymbol{x}}(\cdot) = \sum_{m=1}^{M} F^{m}_{Y|\boldsymbol{X}=\boldsymbol{x}}(\cdot)$$

DCTMs are not enough

Individual deep learning models may make **unreliable predictions**

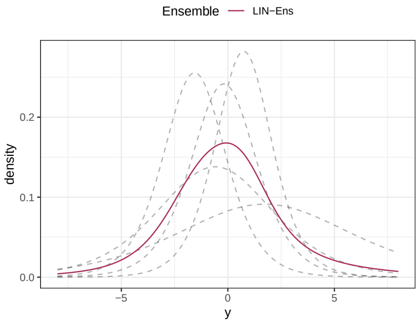
Common criticism:

- No uncertainty quantification
- Small sample sizes
- Stochastic fitting procedure

Deep ensembles improve prediction

1. Fit M instances of the same model
2. Average their M predictions

Deep ensembles lose additivity and interpretability!



$$\bar{F}^M_{Y|\boldsymbol{X}=\boldsymbol{x}}(\cdot) = \sum_{m=1}^{M} F^m_{Y|\boldsymbol{X}=\boldsymbol{x}}(\cdot)$$

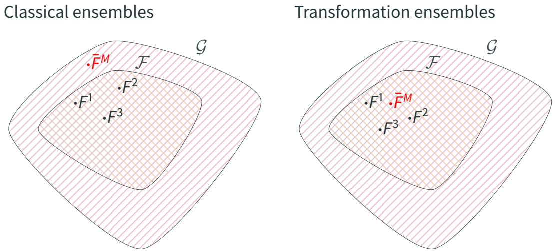
Transformation ensembles

Transformation ensembles average the **transformation function** instead

$$\bar{F}^M_{Y|\boldsymbol{X}=\boldsymbol{x}}(\cdot) := F_Z\left(\sum_{m=1}^{M} h^m_Y(\cdot) + \beta^m(\boldsymbol{x})\right)$$

10.48550/arXiv.2205.12729

- Remain partially **interpretable**



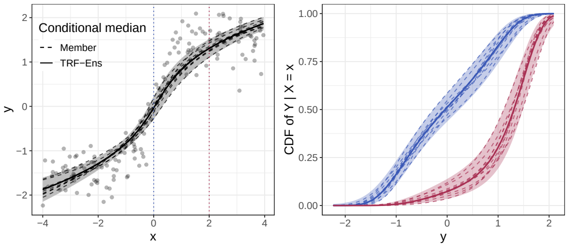
Transformation ensembles

Transformation ensembles average the **transformation function** instead

$$\bar{F}^M_{Y|\boldsymbol{X}=\boldsymbol{x}}(\cdot) := F_Z\left(\sum_{m=1}^{M} h_Y^m(\cdot) + \beta^m(\boldsymbol{x})\right)$$

10.48550/arXiv.2205.12729

- Remain partially **interpretable**
- Quantify **algorithmic uncertainty**



Transformation ensembles

Transformation ensembles average the **transformation function** instead

$$\bar{F}^{M}_{Y|\boldsymbol{X}=\boldsymbol{x}}(\cdot) := F_Z\left(\sum_{m=1}^{M} h_Y^m(\cdot) + \beta^m(\boldsymbol{x})\right)$$

10.48550/arXiv.2205.12729

- Remain partially **interpretable**
- Quantify **algorithmic uncertainty**
- Perform **on par** with deep ensembles

Deep Interpretable Ensembles

Lucas Kook[1,2], Andrea Götschi[1], Philipp F. M. Baumann[3], Torsten Hothorn[1], Beate Sick[1,2]

Fit transformation ensembles in R with ‘deeptrafo’

1. Model formula

```
fm <- vote_count ~ 0 + s(budget, df = 6) +  popularity + deep(texts)
```

Fit transformation ensembles in R with ‘deeptrafo’

1. Model formula

```r
fm <- vote_count ~ 0 + s(budget, df = 6) +  popularity + deep(texts)
```

2. Neural network

```r
embd_mod <- function(x) x |>
  layer_embedding(input_dim = nr_words, output_dim = embedding_size) |>
  layer_lstm(units = 50, return_sequences = TRUE) |>
  layer_lstm(units = 50, return_sequences = FALSE) |> layer_dropout(rate = 0.1) |>
  layer_dense(25) |> layer_dropout(rate = 0.2) |> layer_dense(5) |>
  layer_dropout(rate = 0.3) |> layer_dense(1)
```

Fit transformation ensembles in R with ‘deeptrafo’

1. Model formula

```
fm <- vote_count ~ 0 + s(budget, df = 6) +  popularity + deep(texts)
```

2. Neural network

```
embd_mod <- function(x) x |>
  layer_embedding(input_dim = nr_words, output_dim = embedding_size) |>
  layer_lstm(units = 50, return_sequences = TRUE) |>
  layer_lstm(units = 50, return_sequences = FALSE) |> layer_dropout(rate = 0.1) |>
  layer_dense(25) |> layer_dropout(rate = 0.2) |> layer_dense(5) |>
  layer_dropout(rate = 0.3) |> layer_dense(1)
```

3. Set up model

```
m <- deeptrafo(fm, data = train, list_of_deep_models = list(deep = embd_mod))
```

Fit transformation ensembles in R with 'deeptrafo'

1. Model formula

```
fm <- vote_count ~ 0 + s(budget, df = 6) +  popularity + deep(texts)
```

2. Neural network

```
embd_mod <- function(x) x |>
  layer_embedding(input_dim = nr_words, output_dim = embedding_size) |>
  layer_lstm(units = 50, return_sequences = TRUE) |>
  layer_lstm(units = 50, return_sequences = FALSE) |> layer_dropout(rate = 0.1) |>
  layer_dense(25) |> layer_dropout(rate = 0.2) |> layer_dense(5) |>
  layer_dropout(rate = 0.3) |> layer_dense(1)
```

3. Set up model

```
m <- deeptrafo(fm, data = train, list_of_deep_models = list(deep = embd_mod))
```

4. Fit ensemble

```
ens <- ensemble(m, n_ensemble = 3, epochs = 50, batch_size = 64)
```

Example: Movie ratings

Prediction for a single movie from the test data

- Budget: 2.7×10^8 \$
- Popularity: 57.93
- Overview: “superman returns discover 5 absence allowed lex luthor walk free closest abandoned moved luthor plots ultimate revenge millions killed change planet forever ridding steel ”

Example: Movie ratings

Prediction for a single movie from the test data

- Budget: 2.7×10^8 \$
- Popularity: 57.93
- Overview: “superman returns discover 5 absence allowed lex luthor walk free closest abandoned moved luthor plots ultimate revenge millions killed change planet forever ridding steel ”

Test NLL:

```
unlist(logLik(ens_deep, newdata = test,
  convert_fun = mean))
## members1 members2 members3     mean  ensemble
##     8.24     8.28     8.16     8.23     8.11
```

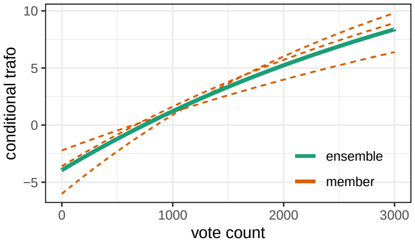
Example: Movie ratings

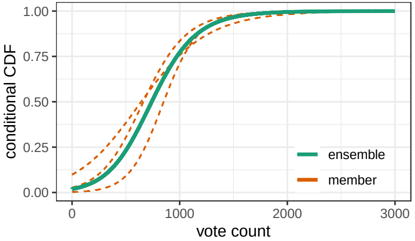
Prediction for a single movie from the test data

- Budget: 2.7×10^8 \$
- Popularity: 57.93
- Overview: “superman returns discover 5 absence allowed lex luthor walk free closest abandoned moved luthor plots ultimate revenge millions killed change planet forever ridding steel ”

Test NLL:

```
unlist(logLik(ens_deep, newdata = test,
  convert_fun = mean))
## members1 members2 members3     mean ensemble
##     8.24     8.28     8.16     8.23     8.11
```





Acknowledgements

PhD Advisors

Beate Sick

Torsten Hothorn

Collaborators

David Rügamer

Oliver Dürr

Philipp FM Baumann

Andrea Götschi

University of Zurich[UZH]

Zurich University of Applied Sciences

zh aw School of Engineering

Swiss National Science Foundation

NOVARTIS