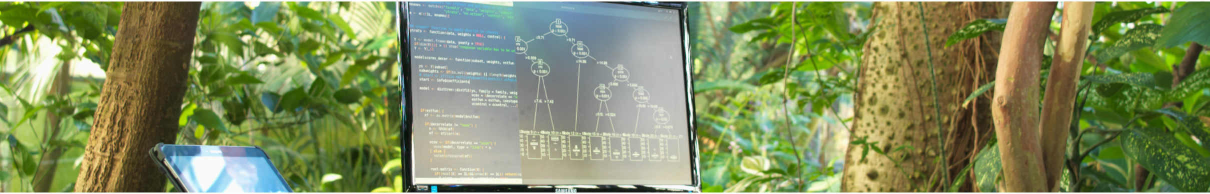
universität innsbruck



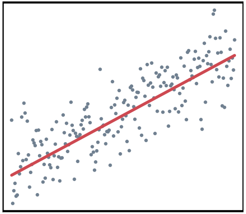
Distributional Regression Forests for Probabilistic Modeling and Forecasting

Achim Zeileis, Lisa Schlosser, Moritz N. Lang, Torsten Hothorn, Georg J. Mayr, Reto Stauffer

http://www.partykit.org/partykit/

Motivation

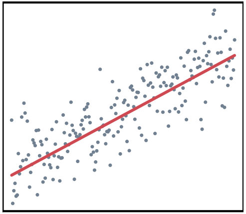
Motivation



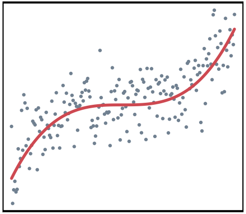
LM, GLM

`lm`
`glm`

Motivation



→



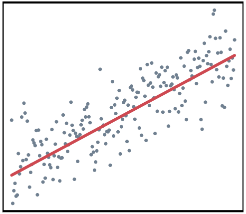
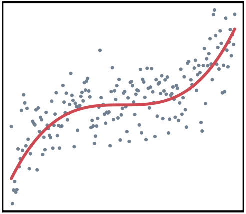
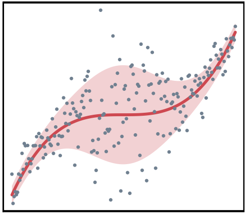
LM, GLM

GAM

`lm`
`glm`

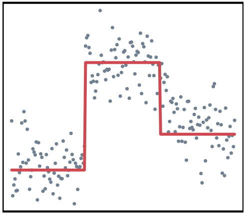
`mgcv`
`VGAM`

Motivation

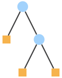
 →  → 

LM, GLM	GAM	GAMLSS
`lm`	`mgcv`	`gamlss`
`glm`	`VGAM`	`mgcv`
		`VGAM`
		`gamboostLSS`
		`bamlss`

Motivation



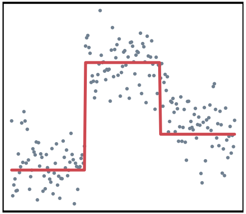
Regression tree

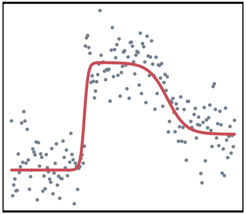


`rpart`

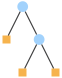
`party(kit)`

Motivation



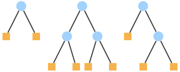


Regression tree



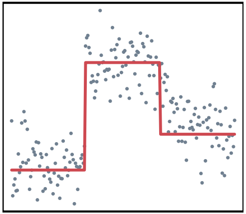
`rpart`
`party(kit)`

Random forest

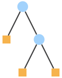


`randomForest`
`ranger`
`party(kit)`

Motivation

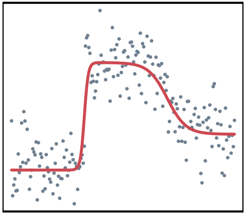


Regression tree

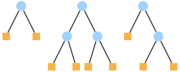


`rpart`
`party(kit)`

→

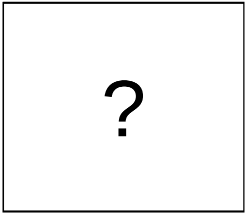


Random forest



`randomForest`
`ranger`
`party(kit)`

→



Distributional trees and forests

`disttree`
based on `partykit`

Motivation

Distributional:

- Specify the complete probability distribution (location, scale, shape, . . .).

Tree:

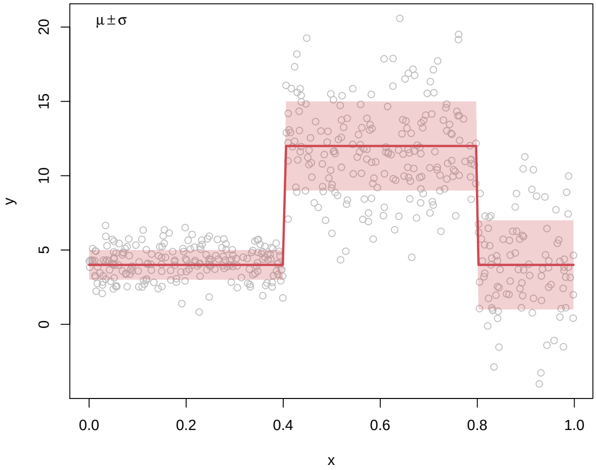
- Automatic detection of steps and abrupt changes.
- Capture non-linear and non-additive effects and interactions.

Forest:

- Smoother effects.
- Stabilization and regularization of the model.

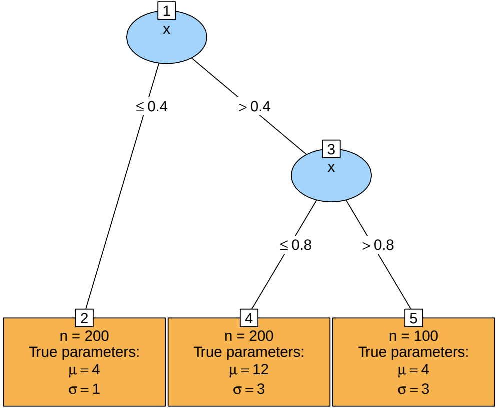
Distributional trees

$$\text{DGP: } Y \mid X = x \sim \mathcal{N}(\mu(x), \sigma^2(x))$$



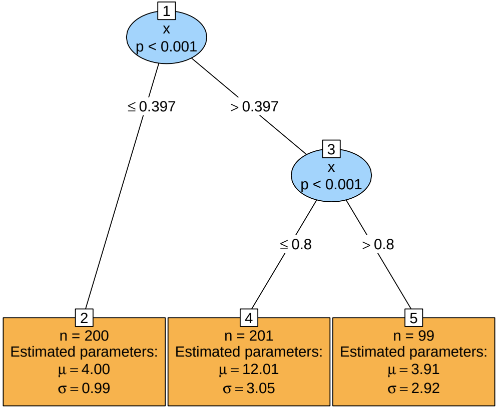
Distributional trees

$$\text{DGP: } Y \mid X = x \sim \mathcal{N}(\mu(x), \sigma^2(x))$$



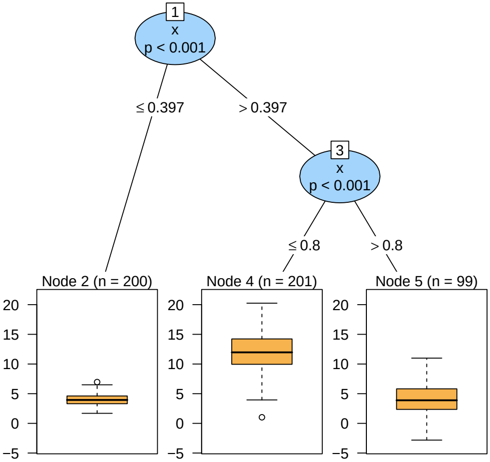
Distributional trees

Model: `disttree(y ~ x)`

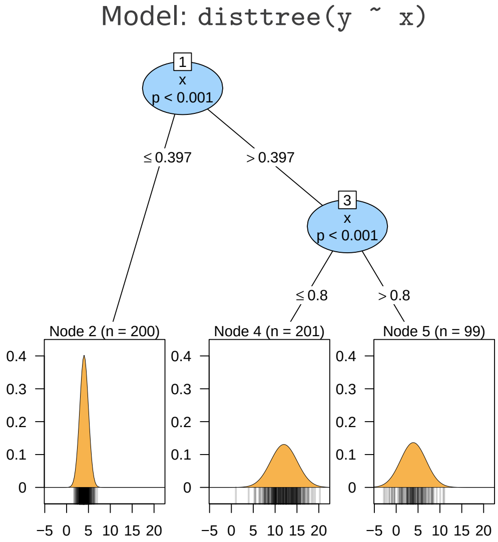


Distributional trees

Model: `disttree(y ~ x)`



Distributional trees

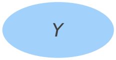


Learning distributional trees and forests

Tree:

Learning distributional trees and forests

Tree:



Learning distributional trees and forests

Tree:

1. Fit global distributional model $\mathcal{D}(Y; \theta)$:
   Estimate $\hat{\theta}$ via maximum likelihood
   $\hat{\theta} = \text{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$

Learning distributional trees and forests

Tree:

1. Fit global distributional model $\mathcal{D}(Y;\theta)$:
   Estimate $\hat{\theta}$ via maximum likelihood
   $\hat{\theta} = \text{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$

$\mathcal{D}(Y;\hat{\theta})$

Learning distributional trees and forests

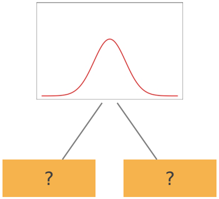
Tree:

1. Fit global distributional model $\mathcal{D}(Y; \theta)$:
   Estimate $\hat{\theta}$ via maximum likelihood
   $\hat{\theta} = \text{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$

Learning distributional trees and forests

Tree:

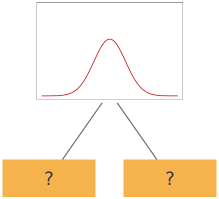
1. Fit global distributional model $\mathcal{D}(Y;\theta)$:
   Estimate $\hat{\theta}$ via maximum likelihood
   $\hat{\theta} = \text{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$



Learning distributional trees and forests

Tree:

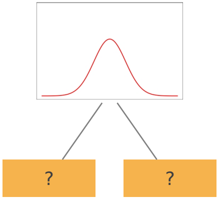
1. Fit global distributional model $\mathcal{D}(Y;\theta)$:
   Estimate $\hat{\theta}$ via maximum likelihood
   $\hat{\theta} = \text{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$
2. Test for associations/instabilities of the scores
   $\frac{\partial \ell}{\partial \theta}(\hat{\theta}; y_i)$ and each covariate X_i.



Learning distributional trees and forests

Tree:

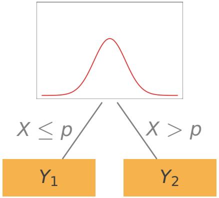
1. Fit global distributional model $\mathcal{D}(Y; \theta)$:
   Estimate $\hat{\theta}$ via maximum likelihood
   $\hat{\theta} = \text{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$
2. Test for associations/instabilities of the scores $\frac{\partial \ell}{\partial \theta}(\hat{\theta}; y_i)$ and each covariate X_j.
3. Split along the covariate X with strongest association or instability and at breakpoint p with highest improvement in log-likelihood.



Learning distributional trees and forests

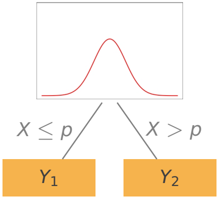
Tree:

1. Fit global distributional model $\mathcal{D}(Y; \theta)$:
   Estimate $\hat{\theta}$ via maximum likelihood
   $\hat{\theta} = \text{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$
2. Test for associations/instabilities of the scores $\frac{\partial \ell}{\partial \theta}(\hat{\theta}; y_i)$ and each covariate X_i.
3. Split along the covariate X with strongest association or instability and at breakpoint p with highest improvement in log-likelihood.



Learning distributional trees and forests

Tree:

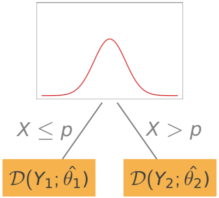


1. Fit global distributional model $\mathcal{D}(Y; \theta)$:
   Estimate $\hat{\theta}$ via maximum likelihood
   $\hat{\theta} = \text{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$
2. Test for associations/instabilities of the scores $\frac{\partial \ell}{\partial \theta}(\hat{\theta}; y_i)$ and each covariate X_j.
3. Split along the covariate X with strongest association or instability and at breakpoint p with highest improvement in log-likelihood.
4. Repeat steps 1–3 recursively until some stopping criterion is met, yielding B subgroups $\mathcal{B}_b$ with $b = 1, \dots, B$.

Learning distributional trees and forests

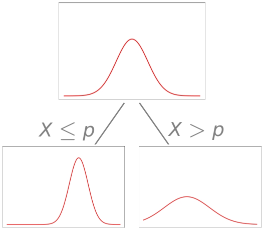
Tree:

1. Fit global distributional model $\mathcal{D}(Y; \theta)$:
   Estimate $\hat{\theta}$ via maximum likelihood
   $\hat{\theta} = \text{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$
2. Test for associations/instabilities of the scores
   $\frac{\partial \ell}{\partial \theta}(\hat{\theta}; y_i)$ and each covariate X_j.
3. Split along the covariate X with strongest association or instability and at breakpoint p with highest improvement in log-likelihood.
4. Repeat steps 1–3 recursively until some stopping criterion is met, yielding B subgroups $\mathcal{B}_b$ with $b = 1, \ldots, B$.



Learning distributional trees and forests

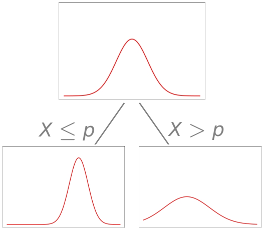
Tree:



1. Fit global distributional model $\mathcal{D}(Y; \theta)$:
   Estimate $\hat{\theta}$ via maximum likelihood
   $\hat{\theta} = \text{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$
2. Test for associations/instabilities of the scores $\frac{\partial \ell}{\partial \theta}(\hat{\theta}; y_i)$ and each covariate X_i.
3. Split along the covariate X with strongest association or instability and at breakpoint p with highest improvement in log-likelihood.
4. Repeat steps 1–3 recursively until some stopping criterion is met, yielding B subgroups $\mathcal{B}_b$ with $b = 1, \dots, B$.

Learning distributional trees and forests

Tree:



1. Fit global distributional model $\mathcal{D}(Y;\theta)$:
   Estimate $\hat{\theta}$ via maximum likelihood
   $\hat{\theta} = \text{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$
2. Test for associations/instabilities of the scores $\frac{\partial \ell}{\partial \theta}(\hat{\theta}; y_i)$ and each covariate X_i.
3. Split along the covariate X with strongest association or instability and at breakpoint p with highest improvement in log-likelihood.
4. Repeat steps 1–3 recursively until some stopping criterion is met, yielding B subgroups $\mathcal{B}_b$ with $b = 1, \dots, B$.

Forest: Ensemble of T trees.

- Bootstrap or subsamples.
- Random input variable sampling.

Adaptive local likelihood estimation

Parameter estimator for **a global** model with learning data $\{y_i\}_{i=1,\ldots,n}$:

$$\hat{\theta} = \operatorname*{argmax}_{\theta\in\Theta} \sum_{i=1}^{n} \ell(\theta; y_i)$$

Adaptive local likelihood estimation

Parameter estimator for **a global** model with learning data $\{(y_i, \mathbf{x}_i)\}_{i=1,\dots,n}$:

$$\hat{\theta}(\mathbf{x}) = \operatorname*{argmax}_{\theta \in \Theta} \sum_{i=1}^{n} w_i(\mathbf{x}) \cdot \ell(\theta; y_i)$$

Adaptive local likelihood estimation

Parameter estimator for **a global** model with learning data $\{(y_i, \mathbf{x}_i)\}_{i=1,\dots,n}$:

$$\hat{\theta}(\mathbf{x}) = \underset{\theta \in \Theta}{\operatorname{argmax}} \sum_{i=1}^{n} w_i(\mathbf{x}) \cdot \ell(\theta; y_i)$$

Weights:

$$w_i^{\text{base}}(\mathbf{x}) \;=\; 1$$

Adaptive local likelihood estimation

Parameter estimator for **an adaptive local** model with learning data $\{(y_i, \mathbf{x}_i)\}_{i=1,\ldots,n}$:

$$\hat{\theta}(\mathbf{x}) = \underset{\theta \in \Theta}{\operatorname{argmax}} \sum_{i=1}^{n} w_i(\mathbf{x}) \cdot \ell(\theta; y_i)$$

Weights:

$$\begin{aligned} w_i^{\text{base}}(\mathbf{x}) &= 1 \\ w_i^{\text{tree}}(\mathbf{x}) &= \sum_{b=1}^{B} I((\mathbf{x}_i \in \mathcal{B}_b) \wedge (\mathbf{x} \in \mathcal{B}_b)) \end{aligned}$$

Adaptive local likelihood estimation

Parameter estimator for **an adaptive local** model with learning data $\{(y_i, \mathbf{x}_i)\}_{i=1,\dots,n}$:

$$\hat{\theta}(\mathbf{x}) = \underset{\theta \in \Theta}{\operatorname{argmax}} \sum_{i=1}^{n} w_i(\mathbf{x}) \cdot \ell(\theta; y_i)$$

Weights:

$$\begin{aligned}
w_i^{\text{base}}(\mathbf{x}) &= 1 \\
w_i^{\text{tree}}(\mathbf{x}) &= \sum_{b=1}^{B} I((\mathbf{x}_i \in \mathcal{B}_b) \wedge (\mathbf{x} \in \mathcal{B}_b)) \\
w_i^{\text{forest}}(\mathbf{x}) &= \frac{1}{T} \sum_{t=1}^{T} \sum_{b=1}^{B^t} I((\mathbf{x}_i \in \mathcal{B}_b^t) \wedge (\mathbf{x} \in \mathcal{B}_b^t))
\end{aligned}$$

Weather forecasting

Goal:

$$X \longrightarrow \boxed{\text{nature}} \longrightarrow Y$$

Data:

- X: State of the atmosphere now (temperature, precipitation, wind, . . .).
- Y: State of the atmosphere in the future (hours, days, weeks, . . .).

Weather forecasting

Goal:

2018-03-15

→ nature →

2018-03-16

Data:

- X: State of the atmosphere now (temperature, precipitation, wind, . . .).
- Y: State of the atmosphere in the future (hours, days, weeks, . . .).

Weather forecasting

Goal:

2018-03-15

$\longrightarrow$ nature $\longrightarrow$

2018-03-16

Two stages:

- Physical model: Numerical weather prediction (NWP).
- Statistical model: Model output statistics (MOS).

Weather forecasting

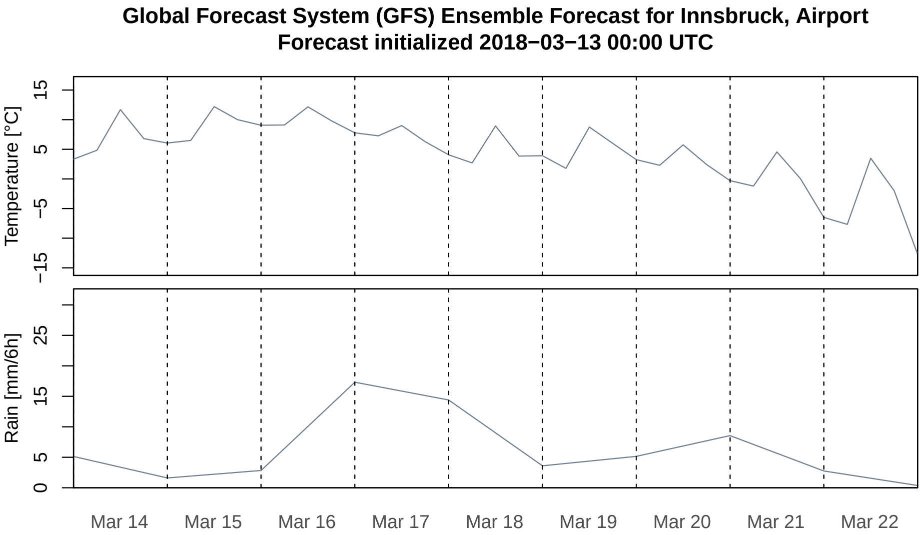
NWP:

- Based on a physical model.
- Massive numerical simulation of atmospheric processes.
- Here: Global model on a $50 \times 50\text{km}^2$ grid.

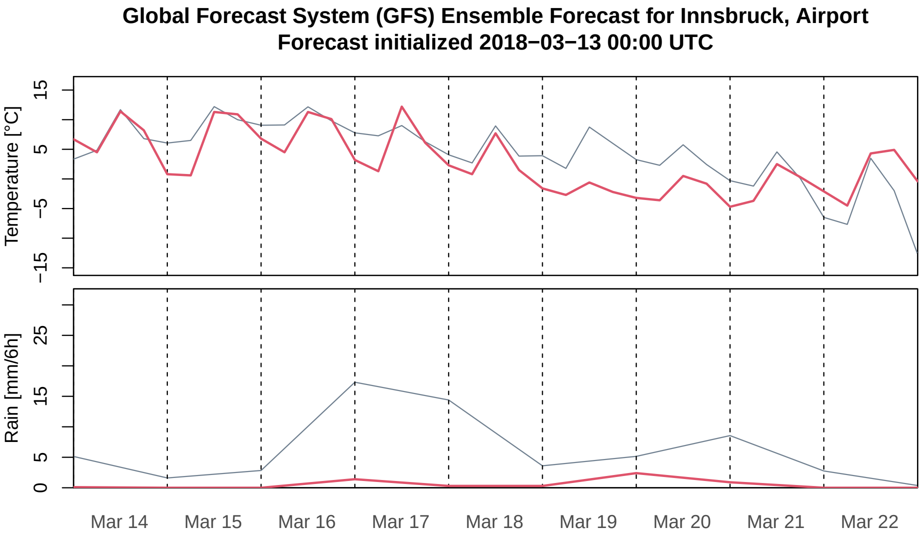
Problem: Uncertain initial conditions, unresolved processes.

Solution: Ensemble of simulation runs under perturbed conditions.

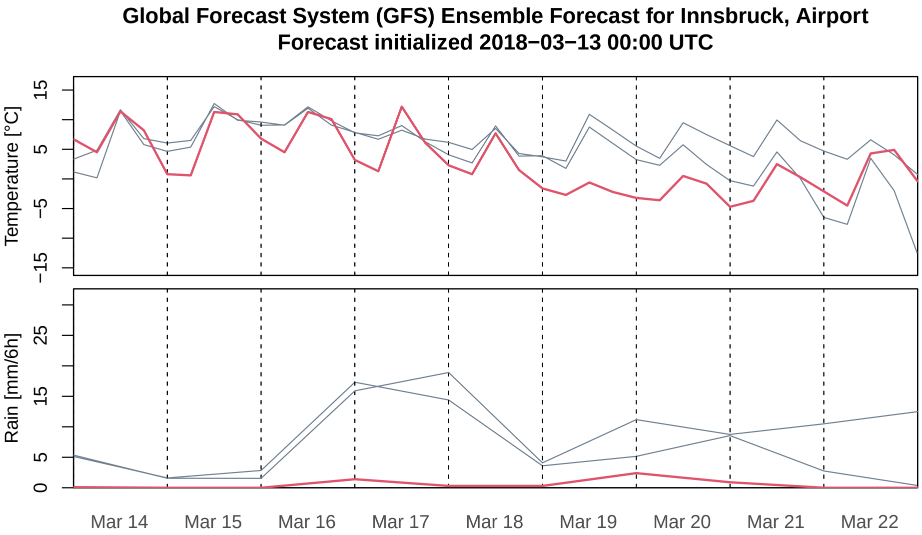
Weather forecasting



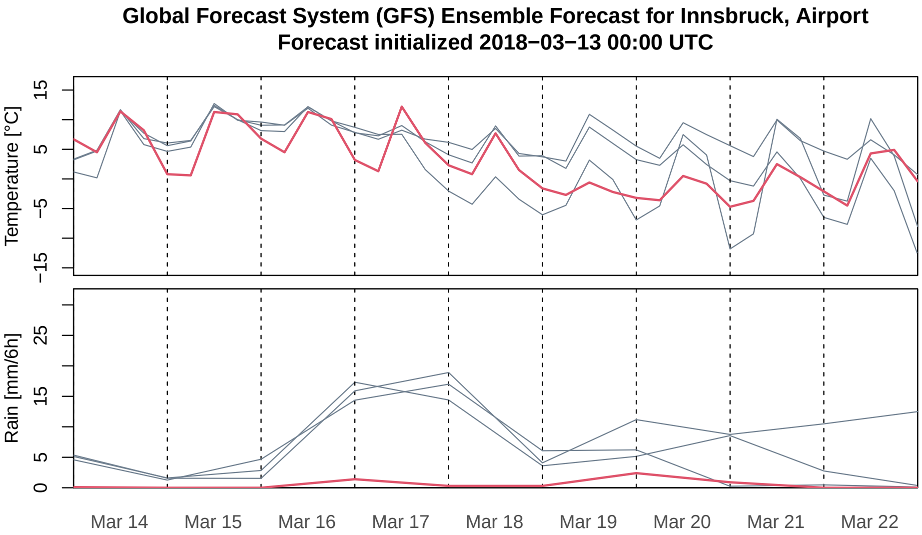
Weather forecasting



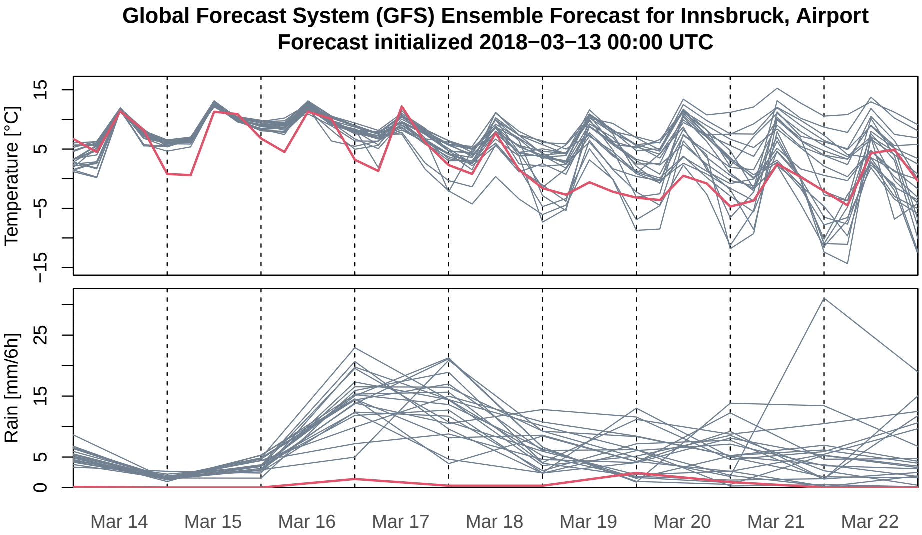
Weather forecasting



Weather forecasting



Weather forecasting



Precipitation forecasting

Goal: Predict daily precipitation amount in complex terrain.

Precipitation forecasting

Goal: Predict daily precipitation amount in complex terrain.

Observation data: National Hydrographical Service.

- Daily 24h precipitation sums from July over 28 years (1985–2012).
- 95 observation stations in Tyrol, Austria.

Precipitation forecasting

Goal: Predict daily precipitation amount in complex terrain.

Observation data: National Hydrographical Service.

- Daily 24h precipitation sums from July over 28 years (1985–2012).
- 95 observation stations in Tyrol, Austria.

NWP: Global Ensemble Forecast System.

- Model outputs: Precipitation, temperature, air pressure, convective available potential energy, downwards short wave radiation flux, . . .
- 80 covariates based on ensemble min/max/mean/standard deviation.

Precipitation forecasting

Goal: Predict daily precipitation amount in complex terrain.

Observation data: National Hydrographical Service.

- Daily 24h precipitation sums from July over 28 years (1985–2012).
- 95 observation stations in Tyrol, Austria.

NWP: Global Ensemble Forecast System.

- Model outputs: Precipitation, temperature, air pressure, convective available potential energy, downwards short wave radiation flux, ...
- 80 covariates based on ensemble min/max/mean/standard deviation.

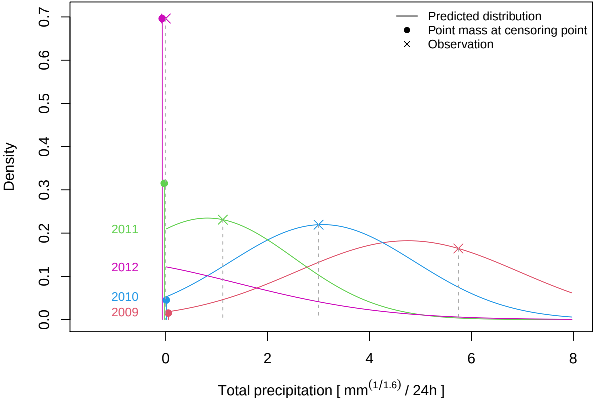
Distribution assumption: Power-transformed Gaussian, censored at 0.

$$(\text{precipitation})^{\frac{1}{1.6}} \sim c\mathcal{N}(\mu, \sigma^2)$$

Precipitation forecasting

Application for one station: Axams.

- Learn forest model on data from 24 years (1985–2008).
- Evaluate on 4 years (2009–2012). Here: July 24.



Precipitation forecasting

Application for one station: Axams.

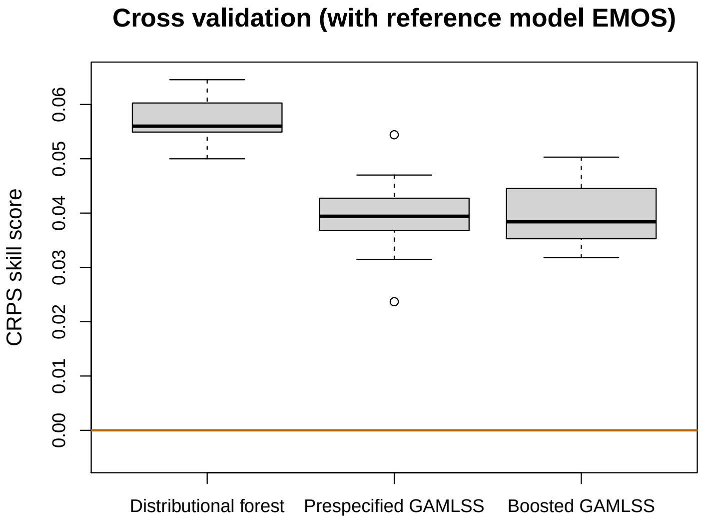
- Learn forest model on data from 24 years.
- Evaluate on 4 years.
- 10 times 7-fold cross validation.

Benchmark: Against other heteroscedastic censored Gaussian models.

- *Ensemble MOS:* Linear predictors using only total precipitation.
- *Prespecified GAMLSS:* Variable selection based on expert knowledge.
- *Boosted GAMLSS:* Automatic variable selection.

Evaluation: Continuous ranked probability skill score.

Precipitation forecasting

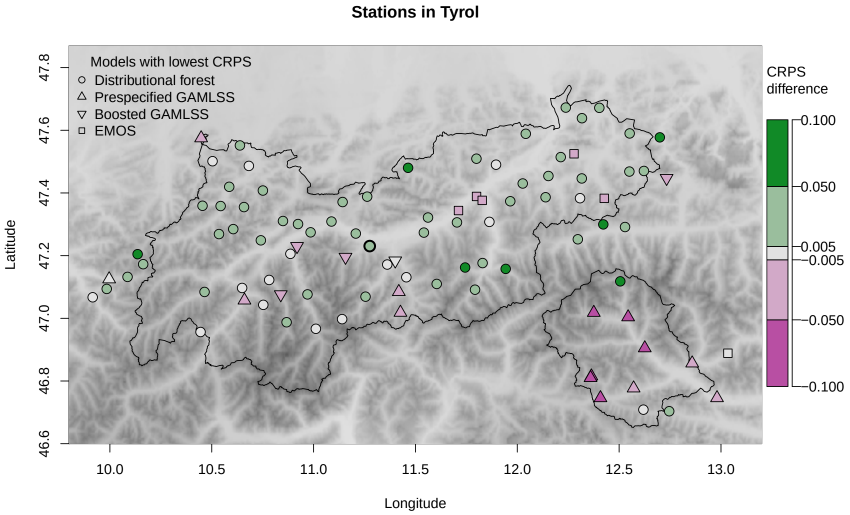


Precipitation forecasting

Application for all 95 stations:

- Learn forest model on data from 24 years (1985–2008).
- Evaluate on 4 years (2009–2012).
- Benchmark against other heteroscedastic censored Gaussian models.

Precipitation forecasting



Wind forecasting

Goal: Nowcasting (1–3 hours ahead) of wind direction at Innsbruck Airport.

Wind forecasting

Goal: Nowcasting (1–3 hours ahead) of wind direction at Innsbruck Airport.

Challenges:

- Circular response in $[0^\circ, 360^\circ)$ with $0^\circ = 360^\circ$.
- Possibly abrupt changes due to geographical position.
- NWP outputs are less useful due to short lead time.

Wind forecasting

Goal: Nowcasting (1–3 hours ahead) of wind direction at Innsbruck Airport.

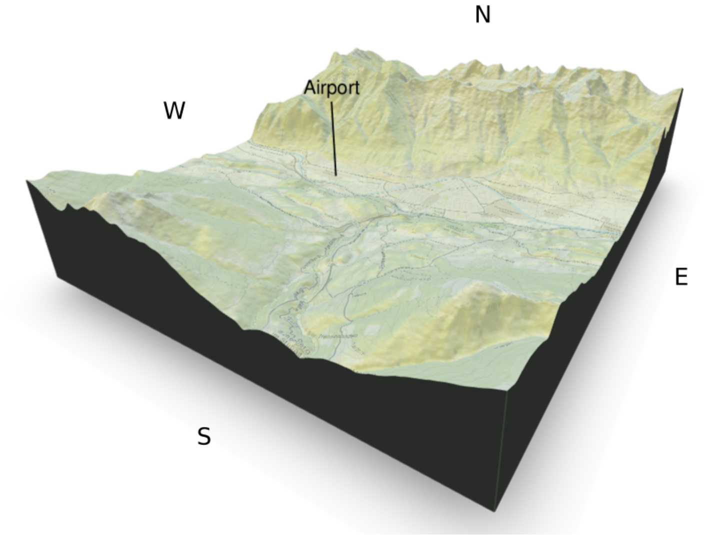
Challenges:

- Circular response in $[0^\circ, 360^\circ)$ with $0^\circ = 360^\circ$.
- Possibly abrupt changes due to geographical position.
- NWP outputs are less useful due to short lead time.

Inputs: Observation data only (41,979 data points).

- 4 stations at Innsbruck Airport, 6 nearby weather stations.
- Base variables: Wind direction, wind (gust) speed, temperature, (reduced) air pressure, relative humidity.
- 260 covariates based on means/minima/maxima, temporal changes, spatial differences towards the airport.

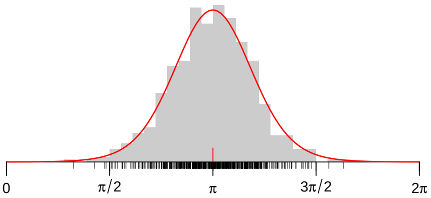
Wind forecasting



Wind forecasting

Distribution assumption: Von Mises.

- Circular normal distribution.
- Location parameter $\mu \in [0, 2\pi)$.
- Concentration parameter $\kappa > 0$.



Log-likelihood: $y \in [0, 2\pi)$ and parameter vector $\theta = (\mu, \kappa)$.

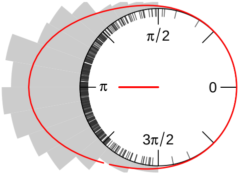
$$\ell(\theta; y) = \log \left\{ \frac{1}{2\pi I_0(\kappa)} \, e^{\kappa \cos(y-\mu)} \right\}$$

where $I_0(\kappa)$ is the modified Bessel function of the first kind and order 0.

Wind forecasting

Distribution assumption: Von Mises.

- Circular normal distribution.
- Location parameter $\mu \in [0, 2\pi)$.
- Concentration parameter $\kappa > 0$.

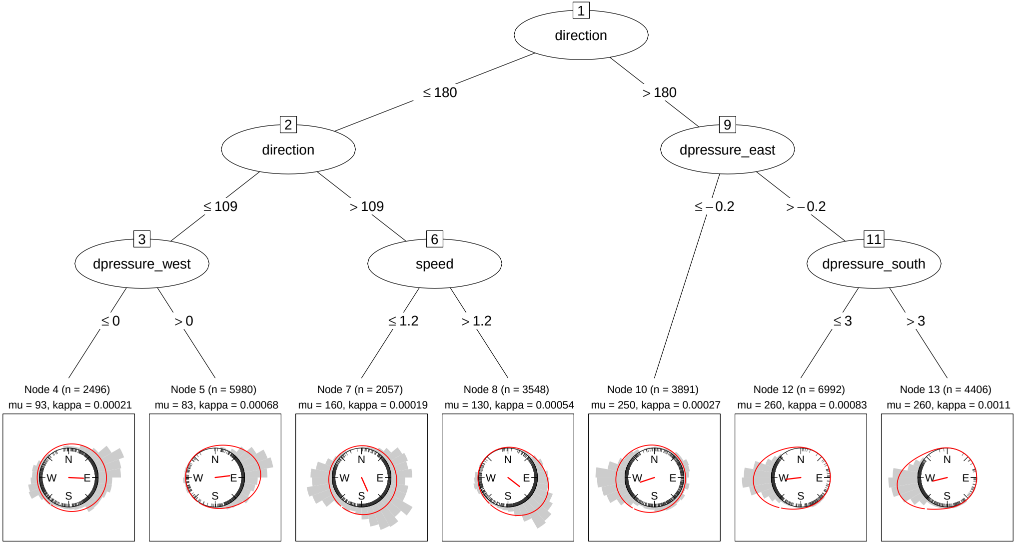


Log-likelihood: $y \in [0, 2\pi)$ and parameter vector $\theta = (\mu, \kappa)$.

$$\ell(\theta; y) = \log \left\{ \frac{1}{2\pi I_0(\kappa)} \, e^{\kappa \cos(y-\mu)} \right\}$$

where $I_0(\kappa)$ is the modified Bessel function of the first kind and order 0.

Wind forecasting



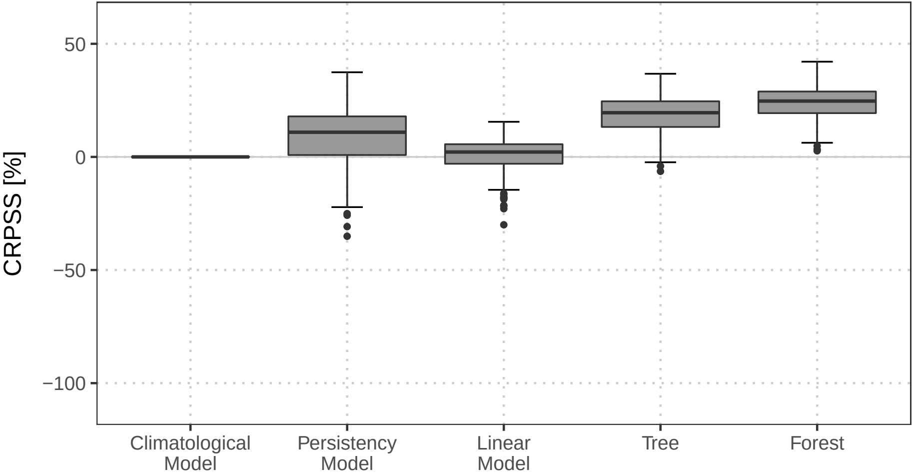
Wind forecasting

Benchmark: Against other naive and circular models.

- Climatology: Without covariates.
- Persistency: Based on current wind direction.
- Circular GLM: Based on current wind speed and wind vectors (u, v).

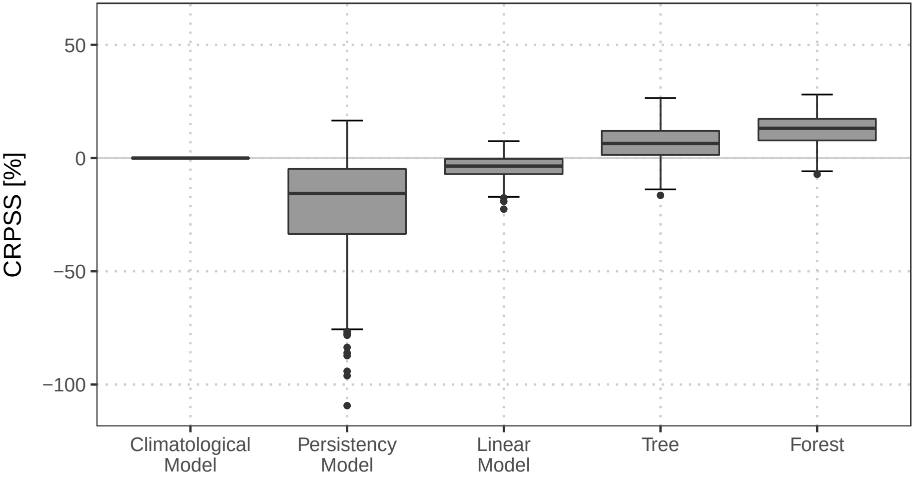
Wind forecasting

Evaluation: CRPS skill score for 1-hourly predictions (5-fold cross validation).



Wind forecasting

Evaluation: CRPS skill score for 3-hourly predictions (5-fold cross validation).



Transformation models

Alternative: When no obvious classic distribution assumption is available.

Advantages:

- Does not require specification of distribution family.
- More flexible framework.

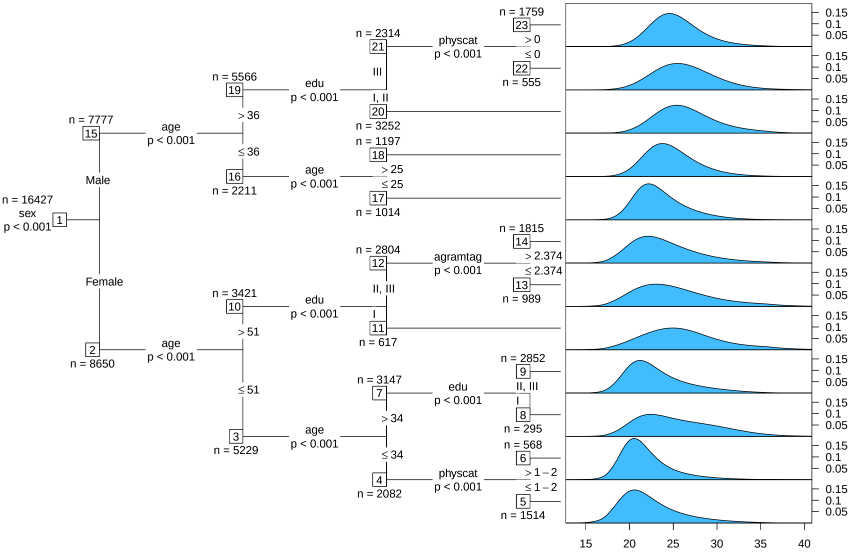
Distribution function:

$$F(y;\theta) = \Phi(\mathbf{a}_{Bs,d}(y)^\top \theta)$$

- $\mathbf{a}_{Bs,d}(y)^\top \theta$ is a smooth, monotone Bernstein polynomial of degree d.
- $d = 1$ corresponds to $\mathcal{N}(\mu, \sigma^2)$.
- $d = 5$ is surprisingly flexible.

Example: Body Mass Index explained by lifestyle factors (Switzerland).

Transformation models



Software

Software: *disttree* and *circtree* available on R-Forge at

`https://R-Forge.R-project.org/projects/partykit/`

Main functions:

`distfit`	Distributional fits (ML, `gamlss.family`/custom `list`).
	No covariates.
`disttree`	Distributional trees (`ctree`/`mob` + `distfit`).
	Covariates as partitioning variables.
`distforest`	Distributional forests (ensemble of `disttrees`).
	Covariates as partitioning variables.

Correspondingly: `circtree`, `circforest`

References

Schlosser L, Hothorn T, Stauffer R, Zeileis A (2019). "Distributional Regression Forests for Probabilistic Precipitation Forecasting in Complex Terrain." *The Annals of Applied Statistics*, **13**(3), 1564–1589. doi:10.1214/19-AOAS1247

Schlosser L, Lang MN, Hothorn T, Mayr GJ, Stauffer R, Zeileis A (2019). "Distributional Trees for Circular Data." *Proceedings of the 34th International Workshop on Statistical Modelling*, **1**, 226–231. https://eeecon.uibk.ac.at/~zeileis/papers/Schlosser+Lang+Hothorn-2019.pdf

Hothorn T, Zeileis A (2017). "Transformation Forests." *arXiv 1701.02110*, arXiv.org E-Print Archive. http://arxiv.org/abs/1701.02110

Hothorn T, Hornik K, Zeileis A (2006). "Unbiased Recursive Partitioning: A Conditional Inference Framework." *Journal of Computational and Graphical Statistics*, **15**(3), 651–674. doi:10.1198/106186006X133933

Zeileis A, Hothorn T, Hornik K (2008). "Model-Based Recursive Partitioning." *Journal of Computational and Graphical Statistics*, **17**(2), 492–514. doi:10.1198/106186008X319331

Hothorn T, Zeileis A (2015). "partykit: A Modular Toolkit for Recursive Partytioning in R." *Journal of Machine Learning Research*, **16**, 3905–3909. http://www.jmlr.org/papers/v16/hothorn15a