Marginally calibrated deep distributional regression

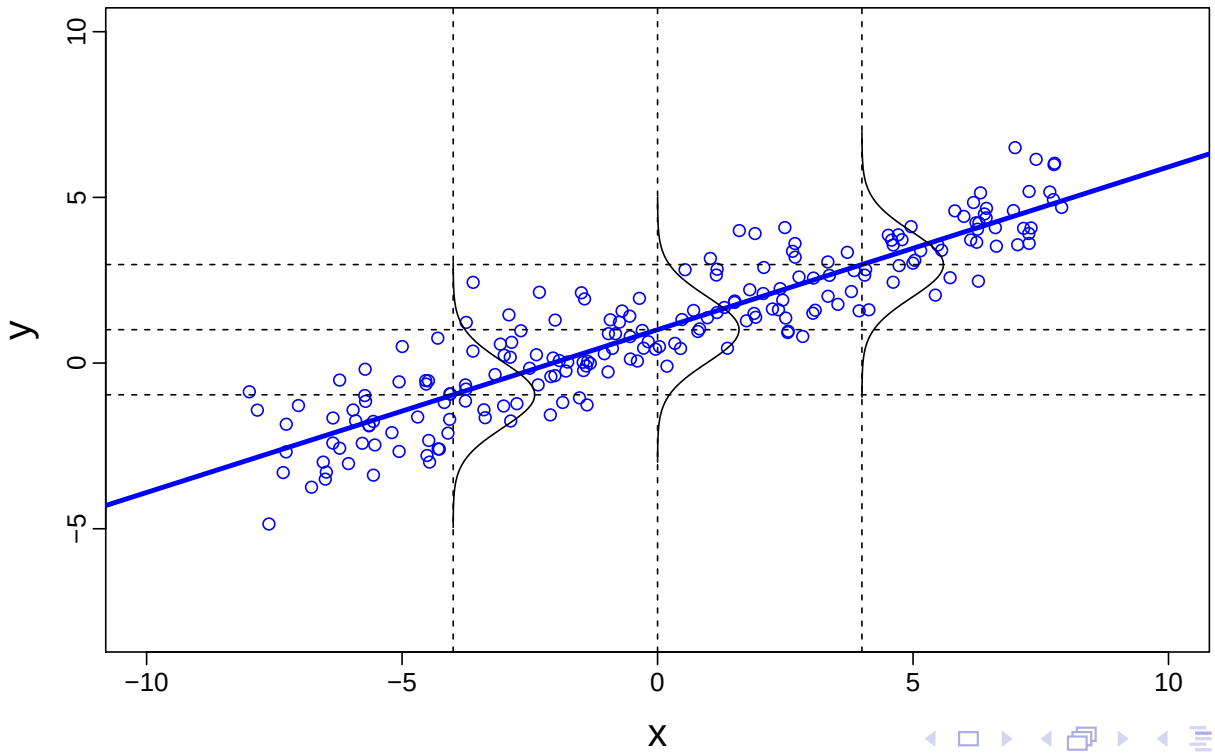
Nadja Klein[1], David Nott[2] and Michael Smith[3]

[1]Humboldt University of Berlin, [2]National University of Singapore and
[3]University of Melbourne

February 18th, 2020

Parametric regression models

- Predict or explain variation in a response variable Y based on features $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p)$.
- We observe a training set of response and feature pairs $(y_i, \boldsymbol{x}_i)$, $i = 1, \ldots, n$.
- A regression model estimates a conditional distribution of Y, given $\boldsymbol{X} = \boldsymbol{x}$, from the training data.



Mean regression

Typically only the mean response is modelled explicitly as a function of $\boldsymbol{x}$

"The ultimate goal of regression analysis is to obtain information about the conditional distribution of a response variable given a set of explanatory variables. This goal is, however, seldom achieved because most established regression models only estimate the conditional mean as a function of the explanatory variables."

[Horthorn *et al.*, 2013]

Distributional regression

Allow aspects of the distribution of y beyond the mean to vary with $\boldsymbol{x}$

- Complex parametric families where higher order moments are $\boldsymbol{x}$-dependent (eg. GAMLSS, Rigby and Stasinopoulos, 2005)
- Bayesian nonparametrics (De Iorio *et al.*, 2004, Foti and Williamson, 2015).
- Mixtures of experts (Jacobs *et al.*, 1991).
- Quantile regression (Koenker and Bassett, 1978).
- Quantile regression forests (Meinshausen, 2006).

Typically, current distributional regression methods are not scalable to either large n or large p.

Distributional regression

Allow aspects of the distribution of y beyond the mean to vary with $\boldsymbol{x}$

- Complex parametric families where higher order moments are $\boldsymbol{x}$-dependent (eg. GAMLSS, Rigby and Stasinopoulos, 2005)
- Bayesian nonparametrics (De Iorio *et al.*, 2004, Foti and Williamson, 2015).
- Mixtures of experts (Jacobs *et al.*, 1991).
- Quantile regression (Koenker and Bassett, 1978).
- Quantile regression forests (Meinshausen, 2006).

Typically, current distributional regression methods are not scalable to either large n or large p.

Distributional regression

Allow aspects of the distribution of y beyond the mean to vary with $\boldsymbol{x}$

- Complex parametric families where higher order moments are $\boldsymbol{x}$-dependent (eg. GAMLSS, Rigby and Stasinopoulos, 2005)
- Bayesian nonparametrics (De Iorio *et al.*, 2004, Foti and Williamson, 2015).
- Mixtures of experts (Jacobs *et al.*, 1991).
- Quantile regression (Koenker and Bassett, 1978).
- Quantile regression forests (Meinshausen, 2006).

Typically, current distributional regression methods are not scalable to either large n or large p.

Distributional regression

Allow aspects of the distribution of y beyond the mean to vary with $\boldsymbol{x}$

- Complex parametric families where higher order moments are $\boldsymbol{x}$-dependent (eg. GAMLSS, Rigby and Stasinopoulos, 2005)
- Bayesian nonparametrics (De Iorio *et al.*, 2004, Foti and Williamson, 2015).
- Mixtures of experts (Jacobs *et al.*, 1991).
- Quantile regression (Koenker and Bassett, 1978).
- Quantile regression forests (Meinshausen, 2006).

Typically, current distributional regression methods are not scalable to either large n or large p.

Distributional regression

Allow aspects of the distribution of y beyond the mean to vary with $\boldsymbol{x}$

- Complex parametric families where higher order moments are $\boldsymbol{x}$-dependent (eg. GAMLSS, Rigby and Stasinopoulos, 2005)
- Bayesian nonparametrics (De Iorio *et al.*, 2004, Foti and Williamson, 2015).
- Mixtures of experts (Jacobs *et al.*, 1991).
- Quantile regression (Koenker and Bassett, 1978).
- Quantile regression forests (Meinshausen, 2006).

Typically, current distributional regression methods are not scalable to either large n or large p.

Distributional regression

Allow aspects of the distribution of y beyond the mean to vary with $\boldsymbol{x}$

- Complex parametric families where higher order moments are $\boldsymbol{x}$-dependent (eg. GAMLSS, Rigby and Stasinopoulos, 2005)
- Bayesian nonparametrics (De Iorio *et al.*, 2004, Foti and Williamson, 2015).
- Mixtures of experts (Jacobs *et al.*, 1991).
- Quantile regression (Koenker and Bassett, 1978).
- Quantile regression forests (Meinshausen, 2006).

Typically, current distributional regression methods are not scalable to either large n or large p.

Distributional regression

Allow aspects of the distribution of y beyond the mean to vary with $\boldsymbol{x}$

- Complex parametric families where higher order moments are $\boldsymbol{x}$-dependent (eg. GAMLSS, Rigby and Stasinopoulos, 2005)
- Bayesian nonparametrics (De Iorio *et al.*, 2004, Foti and Williamson, 2015).
- Mixtures of experts (Jacobs *et al.*, 1991).
- Quantile regression (Koenker and Bassett, 1978).
- Quantile regression forests (Meinshausen, 2006).

Typically, current distributional regression methods are not scalable to either large n or large p.

Mean regression with DNN

- Deep neural network (DNN) - defines a flexible parametrized function $f_\eta(\boldsymbol{x})$ where η denote some learnable parameters (weights).
- For predicting scalar Y from $\boldsymbol{x}$, $f_\eta(\boldsymbol{x})$ will be a scalar-valued function.
- $f_\eta(\boldsymbol{x})$ is a composition of component functions ("layers"): outputs from one layer feed into the next.

Mean regression with DNN

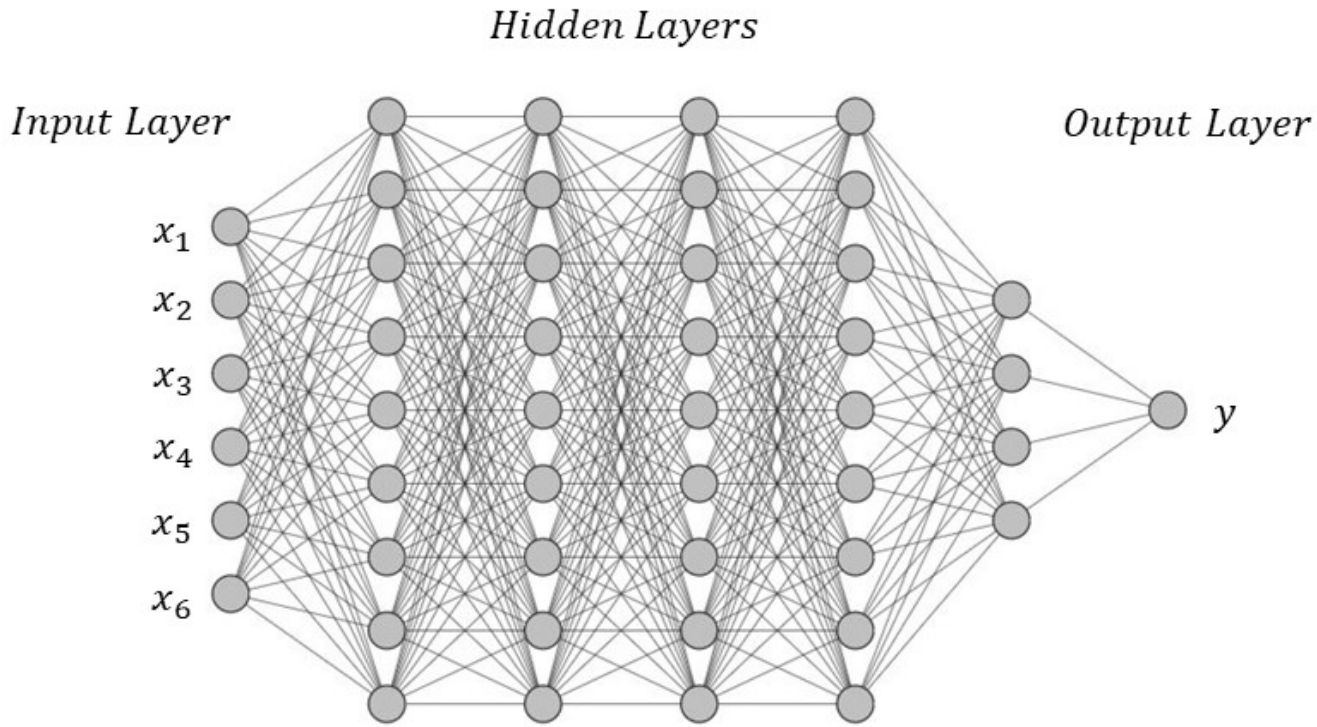
- Deep neural network (DNN) - defines a flexible parametrized function $f_{\eta}(\boldsymbol{x})$ where η denote some learnable parameters (weights).
- For predicting scalar Y from $\boldsymbol{x}$, $f_{\eta}(\boldsymbol{x})$ will be a scalar-valued function.
- $f_{\eta}(\boldsymbol{x})$ is a composition of component functions ("layers"): outputs from one layer feed into the next.

Mean regression with DNN

- Deep neural network (DNN) - defines a flexible parametrized function $f_\eta(\boldsymbol{x})$ where η denote some learnable parameters (weights).
- For predicting scalar Y from $\boldsymbol{x}$, $f_\eta(\boldsymbol{x})$ will be a scalar-valued function.
- $f_\eta(\boldsymbol{x})$ is a composition of component functions ("layers"): outputs from one layer feed into the next.

Deep learning regression model

Simple feed-forward network



Other specialized architectures - convolutional networks (CNN), recurrent networks (RNN), etc.

Training deep learning regression models

- Minimize penalized empirical loss function with respect to η

$$L(\boldsymbol{y}, f_\eta) = \sum_{i=1}^{n} (y_i - f_\eta(\boldsymbol{x}_i))^2$$

- A regularization penalty is usually added to prevent over-fitting
- Regularization can be implicit
- Equivalence to Bayesian posterior mode estimation in a constant variance Gaussian model where $f_\eta(\boldsymbol{x})$ is the conditional mean response.
- For simple uncertainty quantification, a plug-in Gaussian predictive density is often used, after estimating the variance.

Training deep learning regression models

- Minimize penalized empirical loss function with respect to η

$$L(\boldsymbol{y}, f_\eta) = \sum_{i=1}^{n} (y_i - f_\eta(\boldsymbol{x}_i))^2$$

- A regularization penalty is usually added to prevent over-fitting
- Regularization can be implicit
- Equivalence to Bayesian posterior mode estimation in a constant variance Gaussian model where $f_\eta(\boldsymbol{x})$ is the conditional mean response.
- For simple uncertainty quantification, a plug-in Gaussian predictive density is often used, after estimating the variance.

Training deep learning regression models

- Minimize penalized empirical loss function with respect to η

$$L(\boldsymbol{y}, f_\eta) = \sum_{i=1}^{n} (y_i - f_\eta(\boldsymbol{x}_i))^2$$

- A regularization penalty is usually added to prevent over-fitting
- Regularization can be implicit
- Equivalence to Bayesian posterior mode estimation in a constant variance Gaussian model where $f_\eta(\boldsymbol{x})$ is the conditional mean response.
- For simple uncertainty quantification, a plug-in Gaussian predictive density is often used, after estimating the variance.

Training deep learning regression models

- Minimize penalized empirical loss function with respect to η

$$L(\boldsymbol{y}, f_\eta) = \sum_{i=1}^{n} (y_i - f_\eta(\boldsymbol{x}_i))^2$$

- A regularization penalty is usually added to prevent over-fitting
- Regularization can be implicit
- Equivalence to Bayesian posterior mode estimation in a constant variance Gaussian model where $f_\eta(\boldsymbol{x})$ is the conditional mean response.
- For simple uncertainty quantification, a plug-in Gaussian predictive density is often used, after estimating the variance.

Uncertainty quantification in deep learning

Model-based approaches

- Mean-variance networks (eg. Kendall and Gal, 2017)
- Mixture density networks (Bishop, 1994)
- Deep versions of quantile regression (Tagasovska and Lopez-Paz, 2018)

Post-processing adjustments

- Most existing work is for classification, not regression
- Some existing work on probability calibration for regression (Kuleshov *et al.*, 2018)

Postprocessing is computationally cheap, and can be used in a modular way.

Uncertainty quantification in deep learning

Model-based approaches

- Mean-variance networks (eg. Kendall and Gal, 2017)
- Mixture density networks (Bishop, 1994)
- Deep versions of quantile regression (Tagasovska and Lopez-Paz, 2018)

Post-processing adjustments

- Most existing work is for classification, not regression
- Some existing work on probability calibration for regression (Kuleshov *et al.*, 2018)

Postprocessing is computationally cheap, and can be used in a modular way.

Uncertainty quantification in deep learning

Model-based approaches

- Mean-variance networks (eg. Kendall and Gal, 2017)
- Mixture density networks (Bishop, 1994)
- Deep versions of quantile regression (Tagasovska and Lopez-Paz, 2018)

Post-processing adjustments

- Most existing work is for classification, not regression
- Some existing work on probability calibration for regression (Kuleshov *et al.*, 2018)

Postprocessing is computationally cheap, and can be used in a modular way.

Probabilistic forecasting: maximizing sharpness subject to calibration

Murphy and Winkler, 1987, Gneiting *et al.*, 2007

Calibration means that forecasts should be statistically consistent with the observations. There are different kinds of calibration.

- Probability calibration - events given probabililty p by forecaster should occur with relative frequency p.
- Marginal calibration - the average forecast distribution equals the empirical marginal distribution.

An ideal forecast is both probability and marginally calibrated.

Sharpness refers to the concentration of forecasts. Forecasts with less uncertainty are more useful, subject to calibration.

We consider forecasts in the form of regression predictive distributions.

Probabilistic forecasting: maximizing sharpness subject to calibration

Murphy and Winkler, 1987, Gneiting *et al.*, 2007

Calibration means that forecasts should be statistically consistent with the observations. There are different kinds of calibration.

- Probability calibration - events given probabililty p by forecaster should occur with relative frequency p.
- Marginal calibration - the average forecast distribution equals the empirical marginal distribution.

An ideal forecast is both probability and marginally calibrated.

Sharpness refers to the concentration of forecasts. Forecasts with less uncertainty are more useful, subject to calibration.

We consider forecasts in the form of regression predictive distributions.

Probabilistic forecasting: maximizing sharpness subject to calibration

Murphy and Winkler, 1987, Gneiting *et al.*, 2007

Calibration means that forecasts should be statistically consistent with the observations. There are different kinds of calibration.

- Probability calibration - events given probabililty p by forecaster should occur with relative frequency p.
- Marginal calibration - the average forecast distribution equals the empirical marginal distribution.

An ideal forecast is both probability and marginally calibrated.

Sharpness refers to the concentration of forecasts. Forecasts with less uncertainty are more useful, subject to calibration.

We consider forecasts in the form of regression predictive distributions.

Probabilistic forecasting: maximizing sharpness subject to calibration

Murphy and Winkler, 1987, Gneiting *et al.*, 2007

Calibration means that forecasts should be statistically consistent with the observations. There are different kinds of calibration.

- Probability calibration - events given probabililty p by forecaster should occur with relative frequency p.
- Marginal calibration - the average forecast distribution equals the empirical marginal distribution.

An ideal forecast is both probability and marginally calibrated.

Sharpness refers to the concentration of forecasts. Forecasts with less uncertainty are more useful, subject to calibration.

We consider forecasts in the form of regression predictive distributions.

Calibrated uncertainty quantification in deep learning

Kuleshov *et al.* (2018) suggest a postprocessing adjustment of DNN predictive densities to achieve probability calibration, through learning a transformation.

> *"We found that the notion of marginal calibration was too weak for our purposes, since it only preserves guarantees relative to the average distribution"*
>
> Kuleshov *et al.*, 2018

Marginal calibration is a useful complement to probability calibration.

Calibrated uncertainty quantification in deep learning

Kuleshov *et al.* (2018) suggest a postprocessing adjustment of DNN predictive densities to achieve probability calibration, through learning a transformation.

> *"We found that the notion of marginal calibration was too weak for our purposes, since it only preserves guarantees relative to the average distribution"*
>
> Kuleshov *et al.*, 2018

Marginal calibration is a useful complement to probability calibration.

How to achieve marginal calibration?

- A *copula* is a multivariate distribution with marginal distributions uniform on $[0, 1]$.
- *Any multivariate distribution* can be represented by specifying a copula and a set of marginal distributions (a consequence of Sklar's theorem, Sklar, 1959)

Why copulas?

Use of copula representations separates modelling of marginals from modelling dependence structure

- For continuous distributions the copula is unique - if we take a multivariate continuous distribution and extract its copula this is called an 'implicit' or 'inversion' copula.

How to achieve marginal calibration?

- A *copula* is a multivariate distribution with marginal distributions uniform on $[0, 1]$.
- *Any multivariate distribution* can be represented by specifying a copula and a set of marginal distributions (a consequence of Sklar's theorem, Sklar, 1959)

Why copulas?

Use of copula representations separates modelling of marginals from modelling dependence structure

- For continuous distributions the copula is unique - if we take a multivariate continuous distribution and extract its copula this is called an 'implicit' or 'inversion' copula.

How to achieve marginal calibration?

- A *copula* is a multivariate distribution with marginal distributions uniform on $[0, 1]$.
- *Any multivariate distribution* can be represented by specifying a copula and a set of marginal distributions (a consequence of Sklar's theorem, Sklar, 1959)

Why copulas?

Use of copula representations separates modelling of marginals from modelling dependence structure

- For continuous distributions the copula is unique - if we take a multivariate continuous distribution and extract its copula this is called an 'implicit' or 'inversion' copula.

How to achieve marginal calibration?

- A *copula* is a multivariate distribution with marginal distributions uniform on $[0, 1]$.
- *Any multivariate distribution* can be represented by specifying a copula and a set of marginal distributions (a consequence of Sklar's theorem, Sklar, 1959)

Why copulas?

Use of copula representations separates modelling of marginals from modelling dependence structure

- For continuous distributions the copula is unique - if we take a multivariate continuous distribution and extract its copula this is called an 'implicit' or 'inversion' copula.

Copula model

Consider observations $Y = (Y_1, \ldots, Y_n)^\top$ of a continuous response, with feature values $\boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$

- $$p(\boldsymbol{y}|\boldsymbol{x}) = c^\dagger(F(y_1|\boldsymbol{x}_1), \ldots, F(y_n|\boldsymbol{x}_n)|\boldsymbol{x}) \prod_{i=1}^{n} p(y_i|\boldsymbol{x}_i)$$
- $c^\dagger$ density of a parametric copula with parameters θ: we use an implicit copula of a deep neural network regression model, $c_{\text{DNN}}(\boldsymbol{u}|\boldsymbol{x}, \theta)$
- Assume the distribution of $Y_i|\boldsymbol{x}_i$ has an invariant margin, so that $p(y_i|\boldsymbol{x}_i) = p_Y(y_i)$ with distribution function F_Y
- $$p(\boldsymbol{y}|\boldsymbol{x}, \theta) = c_{\text{DNN}}(F_Y(y_1), \ldots, F_Y(y_n)|\boldsymbol{x}, \theta) \prod_{i=1}^{n} p_Y(y_i)$$

Copula model

Consider observations $Y = (Y_1, \ldots, Y_n)^\top$ of a continuous response, with feature values $\boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$

- $$p(\boldsymbol{y}|\boldsymbol{x}) = c^\dagger(F(y_1|\boldsymbol{x}_1), \ldots, F(y_n|\boldsymbol{x}_n)|\boldsymbol{x}) \prod_{i=1}^{n} p(y_i|\boldsymbol{x}_i)$$
- $c^\dagger$ density of a parametric copula with parameters θ: we use an implicit copula of a deep neural network regression model, $c_{\text{DNN}}(\boldsymbol{u}|\boldsymbol{x}, \theta)$
- Assume the distribution of $Y_i|\boldsymbol{x}_i$ has an invariant margin, so that $p(y_i|\boldsymbol{x}_i) = p_Y(y_i)$ with distribution function F_Y
- $$p(\boldsymbol{y}|\boldsymbol{x}, \theta) = c_{\text{DNN}}(F_Y(y_1), \ldots, F_Y(y_n)|\boldsymbol{x}, \theta) \prod_{i=1}^{n} p_Y(y_i)$$

Copula model

Consider observations $Y = (Y_1, \ldots, Y_n)^\top$ of a continuous response, with feature values $\boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$

- $$p(\boldsymbol{y}|\boldsymbol{x}) = c^\dagger(F(y_1|\boldsymbol{x}_1), \ldots, F(y_n|\boldsymbol{x}_n)|\boldsymbol{x}) \prod_{i=1}^{n} p(y_i|\boldsymbol{x}_i)$$
- $c^\dagger$ density of a parametric copula with parameters θ: we use an implicit copula of a deep neural network regression model, $c_{\text{DNN}}(\boldsymbol{u}|\boldsymbol{x}, \theta)$
- Assume the distribution of $Y_i|\boldsymbol{x}_i$ has an invariant margin, so that $p(y_i|\boldsymbol{x}_i) = p_Y(y_i)$ with distribution function F_Y
- $$p(\boldsymbol{y}|\boldsymbol{x}, \theta) = c_{\text{DNN}}(F_Y(y_1), \ldots, F_Y(y_n)|\boldsymbol{x}, \theta) \prod_{i=1}^{n} p_Y(y_i)$$

Copula model

Consider observations $Y = (Y_1, \ldots, Y_n)^\top$ of a continuous response, with feature values $\boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$

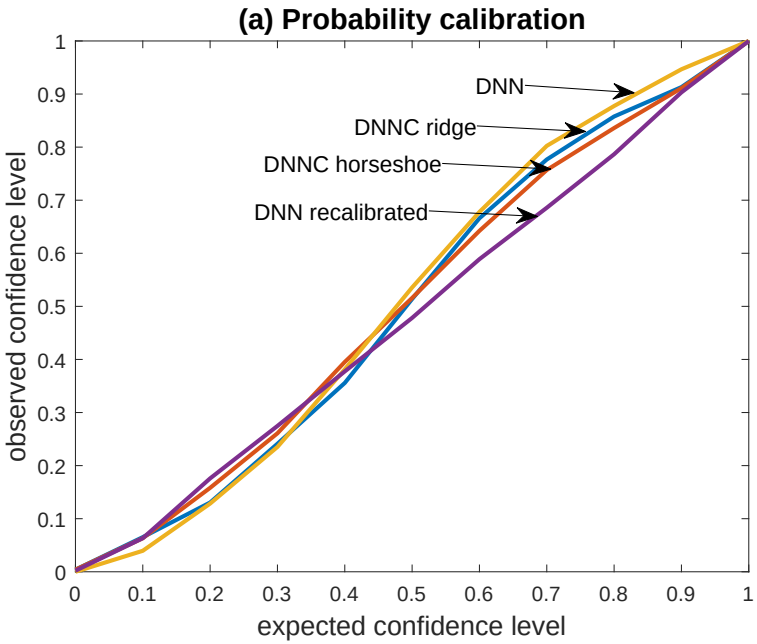
- $$p(\boldsymbol{y}|\boldsymbol{x}) = c^\dagger(F(y_1|\boldsymbol{x}_1), \ldots, F(y_n|\boldsymbol{x}_n)|\boldsymbol{x}) \prod_{i=1}^{n} p(y_i|\boldsymbol{x}_i)$$
- $c^\dagger$ density of a parametric copula with parameters θ: we use an implicit copula of a deep neural network regression model, $c_{\text{DNN}}(\boldsymbol{u}|\boldsymbol{x}, \theta)$
- Assume the distribution of $Y_i|\boldsymbol{x}_i$ has an invariant margin, so that $p(y_i|\boldsymbol{x}_i) = p_Y(y_i)$ with distribution function F_Y
- $$p(\boldsymbol{y}|\boldsymbol{x}, \theta) = c_{\text{DNN}}(F_Y(y_1), \ldots, F_Y(y_n)|\boldsymbol{x}, \theta) \prod_{i=1}^{n} p_Y(y_i)$$

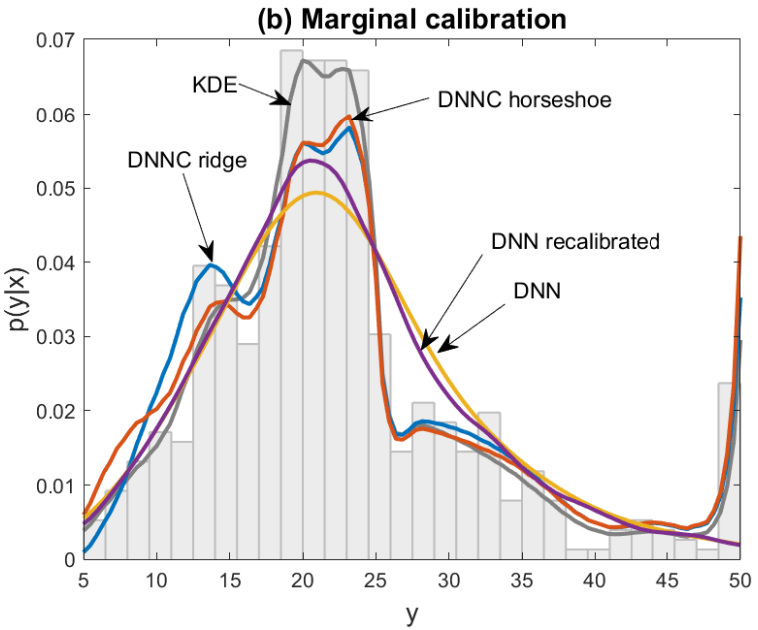
Boston housing data

Calibration

- Response is house price
- 13 features relating to geographical information, crime rate, air pollution, schools, number of rooms, etc.
- The response has a marginal distribution with very complex features
- A DNN copula regression provides superior out-of-sample prediction to an uncalibrated DNN and DNN with probability recalibration (Kuleshov *et al.*, 2018)

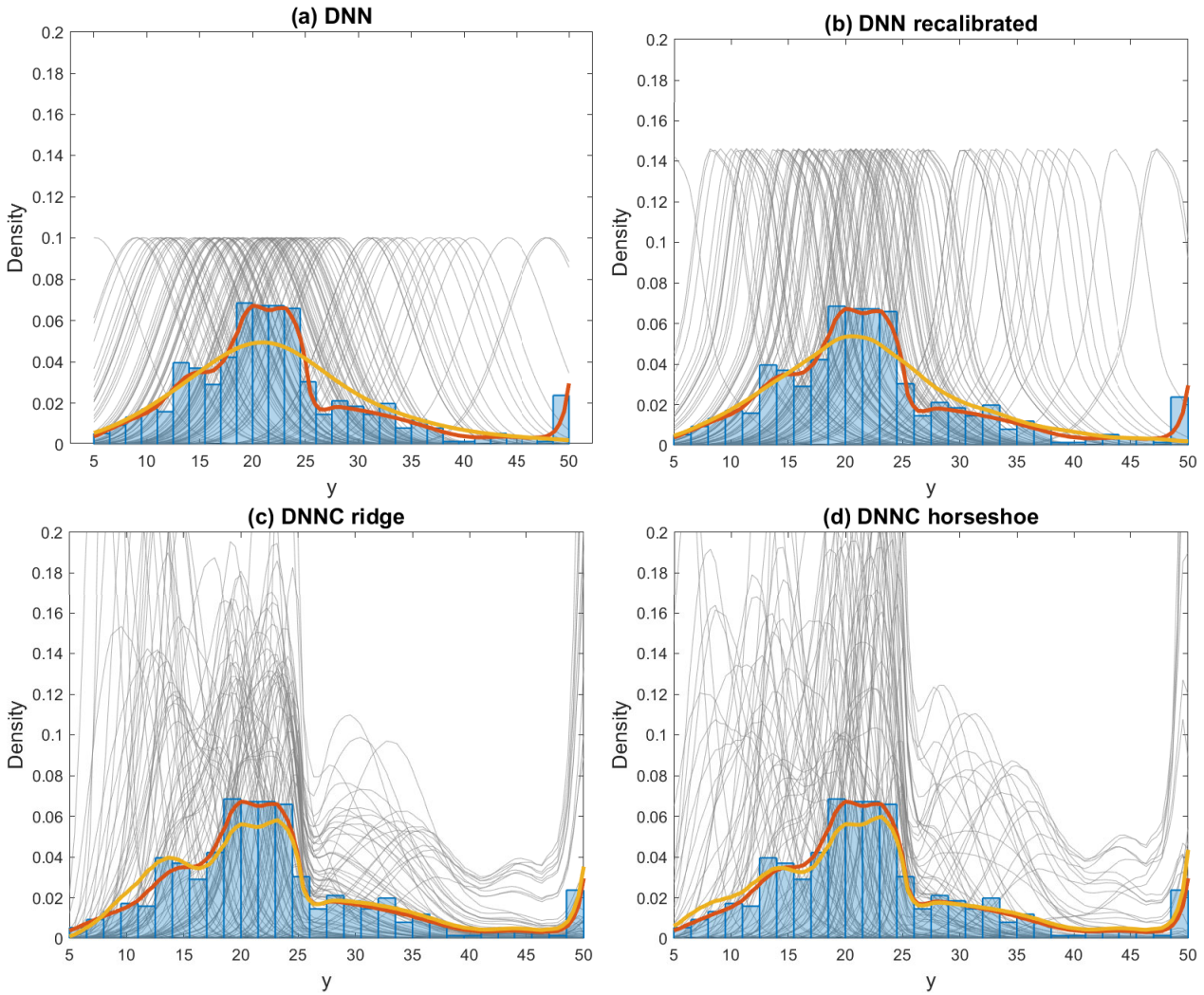
Boston housing data





Boston housing data

Predictive densities



Application to (Bayesian) likelihood-free inference

- Let ρ denote the parameters in a parametric statistical model for data $\boldsymbol{d}$ with density $p(\boldsymbol{d}|\rho)$

Bayesian inference

Set up a full probability model for $(\rho, \boldsymbol{d})$ and then condition on the observed data $\boldsymbol{d}_{\text{obs}}$ to learn about ρ

- Full probability model: $p(\rho, \boldsymbol{d}) = p(\rho)p(\boldsymbol{d}|\rho)$, where $p(\rho)$ is a prior density expressing what is known before observing data.
- For inference, use the *posterior density* $p(\rho|\boldsymbol{d}_{\text{obs}})$.

Application to (Bayesian) likelihood-free inference

- Let ρ denote the parameters in a parametric statistical model for data $\boldsymbol{d}$ with density $p(\boldsymbol{d}|\rho)$

Bayesian inference

Set up a full probability model for $(\rho, \boldsymbol{d})$ and then condition on the observed data $\boldsymbol{d}_{\text{obs}}$ to learn about ρ

- Full probability model: $p(\rho, \boldsymbol{d}) = p(\rho)p(\boldsymbol{d}|\rho)$, where $p(\rho)$ is a prior density expressing what is known before observing data.
- For inference, use the *posterior density* $p(\rho|\boldsymbol{d}_{\text{obs}})$.

Application to (Bayesian) likelihood-free inference

- Let ρ denote the parameters in a parametric statistical model for data $\boldsymbol{d}$ with density $p(\boldsymbol{d}|\rho)$

Bayesian inference

Set up a full probability model for $(\rho, \boldsymbol{d})$ and then condition on the observed data $\boldsymbol{d}_{\text{obs}}$ to learn about ρ

- Full probability model: $p(\rho, \boldsymbol{d}) = p(\rho)p(\boldsymbol{d}|\rho)$, where $p(\rho)$ is a prior density expressing what is known before observing data.
- For inference, use the *posterior density* $p(\rho|\boldsymbol{d}_{\text{obs}})$.

Application to (Bayesian) likelihood-free inference

- Let ρ denote the parameters in a parametric statistical model for data $\boldsymbol{d}$ with density $p(\boldsymbol{d}|\rho)$

Bayesian inference

Set up a full probability model for $(\rho, \boldsymbol{d})$ and then condition on the observed data $\boldsymbol{d}_{\text{obs}}$ to learn about ρ

- Full probability model: $p(\rho, \boldsymbol{d}) = p(\rho)p(\boldsymbol{d}|\rho)$, where $p(\rho)$ is a prior density expressing what is known before observing data.
- For inference, use the *posterior density* $p(\rho|\boldsymbol{d}_{\text{obs}})$.

Application to (Bayesian) likelihood-free inference

- Frequently, interesting models are specified in terms of how you generate data from them rather through an explicit form for $p(\boldsymbol{d}|\rho)$.

Likelihood-free Bayesian computation by regression

- Simulate data $(\rho_i, \boldsymbol{d}_i) \sim p(\rho)p(\boldsymbol{d}|\rho)$, $i = 1, \ldots, n$,
- Fit a distributional regression model with ρ_i as response and $\boldsymbol{d}_i$ as features,
- The predictive distribution from the regression at $\boldsymbol{d} = \boldsymbol{d}_{\text{obs}}$ is an esitmate of $p(\rho|\boldsymbol{d}_{\text{obs}})$.

- We don't need to compute $p(\boldsymbol{d}|\rho)$ anywhere.
- We use our regression copula approach for the distributional regression. Motivation: the true posterior is marginally calibrated.

Application to (Bayesian) likelihood-free inference

- Frequently, interesting models are specified in terms of how you generate data from them rather through an explicit form for $p(\boldsymbol{d}|\rho)$.

Likelihood-free Bayesian computation by regression

- Simulate data $(\rho_i, \boldsymbol{d}_i) \sim p(\rho)p(\boldsymbol{d}|\rho)$, $i = 1, \ldots, n$,
- Fit a distributional regression model with ρ_i as response and $\boldsymbol{d}_i$ as features,
- The predictive distribution from the regression at $\boldsymbol{d} = \boldsymbol{d}_{\text{obs}}$ is an esitmate of $p(\rho|\boldsymbol{d}_{\text{obs}})$.

- We don't need to compute $p(\boldsymbol{d}|\rho)$ anywhere.
- We use our regression copula approach for the distributional regression. Motivation: the true posterior is marginally calibrated.

Application to (Bayesian) likelihood-free inference

- Frequently, interesting models are specified in terms of how you generate data from them rather through an explicit form for $p(\boldsymbol{d}|\rho)$.

Likelihood-free Bayesian computation by regression

- Simulate data $(\rho_i, \boldsymbol{d}_i) \sim p(\rho)p(\boldsymbol{d}|\rho)$, $i = 1, \ldots, n$,
- Fit a distributional regression model with ρ_i as response and $\boldsymbol{d}_i$ as features,
- The predictive distribution from the regression at $\boldsymbol{d} = \boldsymbol{d}_{\text{obs}}$ is an esitmate of $p(\rho|\boldsymbol{d}_{\text{obs}})$.

- We don't need to compute $p(\boldsymbol{d}|\rho)$ anywhere.
- We use our regression copula approach for the distributional regression. Motivation: the true posterior is marginally calibrated.

Application to (Bayesian) likelihood-free inference

- Frequently, interesting models are specified in terms of how you generate data from them rather through an explicit form for $p(\boldsymbol{d}|\boldsymbol{\rho})$.

Likelihood-free Bayesian computation by regression

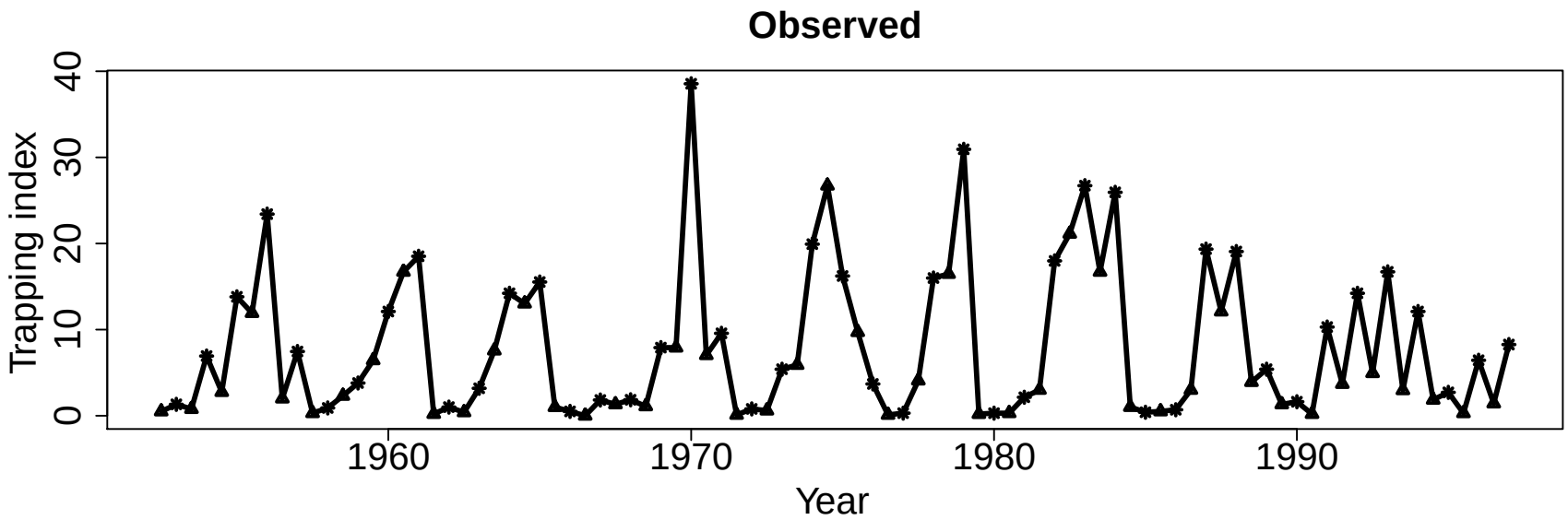
- Simulate data $(\boldsymbol{\rho}_i, \boldsymbol{d}_i) \sim p(\boldsymbol{\rho})p(\boldsymbol{d}|\boldsymbol{\rho})$, $i = 1, \ldots, n$,
- Fit a distributional regression model with $\boldsymbol{\rho}_i$ as response and $\boldsymbol{d}_i$ as features,
- The predictive distribution from the regression at $\boldsymbol{d} = \boldsymbol{d}_{\text{obs}}$ is an esitmate of $p(\boldsymbol{\rho}|\boldsymbol{d}_{\text{obs}})$.

- We don't need to compute $p(\boldsymbol{d}|\boldsymbol{\rho})$ anywhere.
- We use our regression copula approach for the distributional regression. Motivation: the true posterior is marginally calibrated.

Voles data

Turchin and Ellner, 2000, Fasiolo and Wood, 2018

- Trapping data on Voles abundance from Kilpisjarvi, Finland.
- 90 data points collected during spring and autumn of each year, 1952-1997



Voles data

Turchin and Ellner, 2000, Fasiolo and Wood, 2018

Continuous time stochastic differential equation model for scaled abundances (n_t, p_t), $t \geq 0$,of Fennoscandian voles and weasels.

Parameters

- r, s - intrinsic growth rates for voles and weasels
- e seasonal modulation parameter
- g, h - maximal rate of mortality inflicted by generalist predators and half-saturation parameter, respectively
- a, δ - maximal predation rate for individual weasels, half saturation prey density
- ϕ sampling rate parameter, $d_t \sim \text{Poisson}(\phi n_t)$.
- σ - standard deviation of driving Brownian motion

Convolutional neural networks

- We use an implicit copula of a convolutional network to predict the components of the parameter vector $\boldsymbol{\rho} \in \mathbb{R}^p$ based on data $\boldsymbol{d} = (\boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_T^\top)^\top$
- Can be estimated based on training sets of simulations of pairs $(\boldsymbol{\rho}_k, \boldsymbol{d}_k)$ which are generated from the joint model
- Does not require manual specification of 'summary statistics'
- 10,000 data sets simulated under the prior, and use 8,000 data sets for training and 2,000 for testing.

Convolutional neural networks

- We use an implicit copula of a convolutional network to predict the components of the parameter vector $\boldsymbol{\rho} \in \mathbb{R}^p$ based on data $\boldsymbol{d} = (\boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_T^\top)^\top$
- Can be estimated based on training sets of simulations of pairs $(\boldsymbol{\rho}_k, \boldsymbol{d}_k)$ which are generated from the joint model
- Does not require manual specification of 'summary statistics'
- 10,000 data sets simulated under the prior, and use 8,000 data sets for training and 2,000 for testing.

Convolutional neural networks

- We use an implicit copula of a convolutional network to predict the components of the parameter vector $\boldsymbol{\rho} \in \mathbb{R}^p$ based on data $\boldsymbol{d} = (\boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_T^\top)^\top$
- Can be estimated based on training sets of simulations of pairs $(\boldsymbol{\rho}_k, \boldsymbol{d}_k)$ which are generated from the joint model
- Does not require manual specification of 'summary statistics'
- 10,000 data sets simulated under the prior, and use 8,000 data sets for training and 2,000 for testing.

Convolutional neural networks

- We use an implicit copula of a convolutional network to predict the components of the parameter vector $\boldsymbol{\rho} \in \mathbb{R}^p$ based on data $\boldsymbol{d} = (\boldsymbol{d}_1^\top, \ldots, \boldsymbol{d}_T^\top)^\top$
- Can be estimated based on training sets of simulations of pairs $(\boldsymbol{\rho}_k, \boldsymbol{d}_k)$ which are generated from the joint model
- Does not require manual specification of 'summary statistics'
- 10,000 data sets simulated under the prior, and use 8,000 data sets for training and 2,000 for testing.

Benchmark study for the DNNC

(i) DNN as implemented in the `keras` R package (Chollet and Allaire, 2018)

(ii) ABC as implemented in the R package `abc` (Csilléry *et al.*, 2012)

(iii) ABCrf as implemented in the R package `abcrf` (Marin *et al.*, 2017)

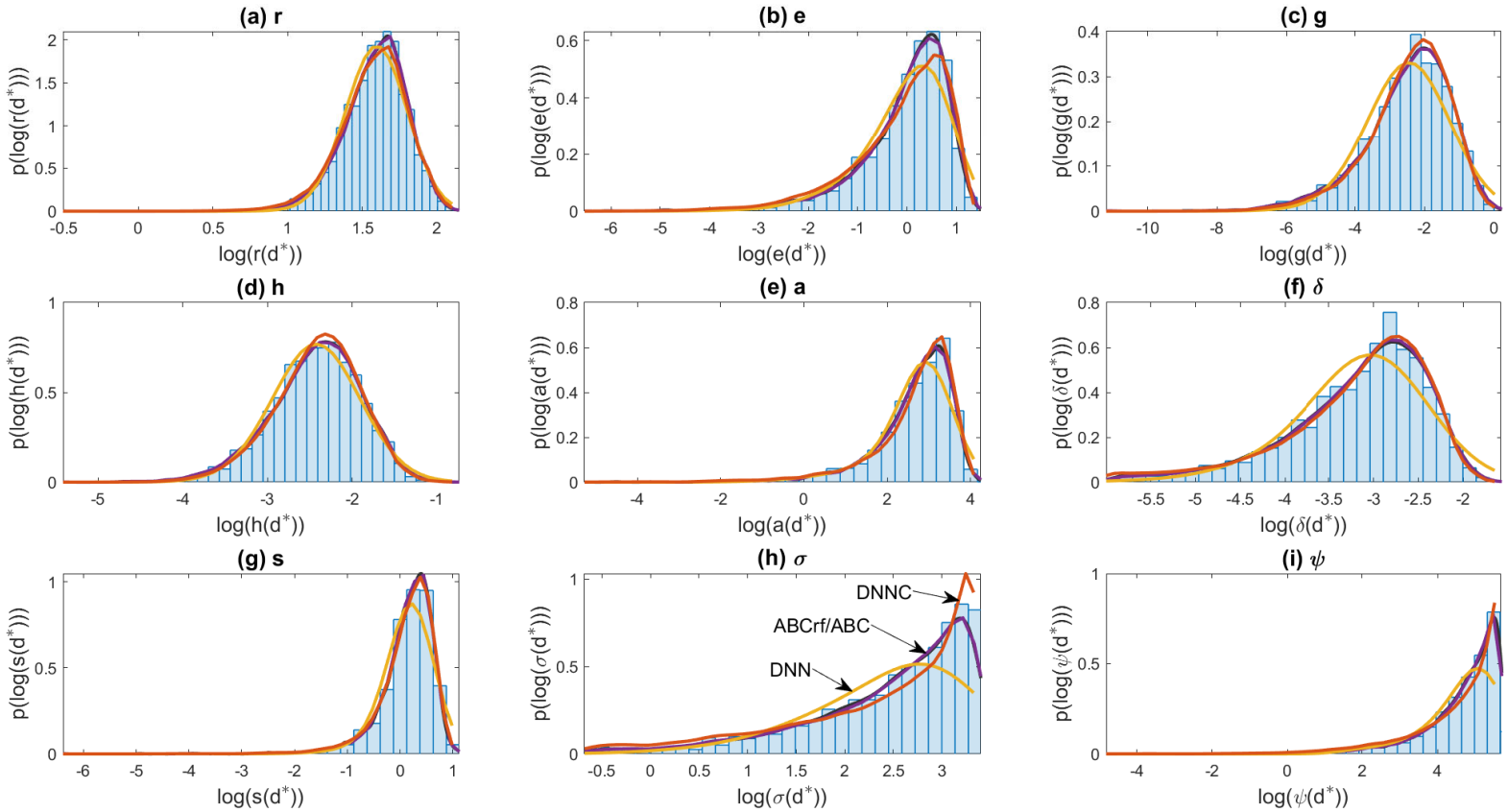
(iv) BSL the Bayesian synthetic likelihood approach implemented in the R package `BSL` (Price *et al.*, 2019).

(v) semiBSL a semi-parametric version of BSL

Results

- DNNC generally outperforms other benchmarks, with smallest simulation MSE values, and with coverage rates closest to the nominal levels
- ABC, DNNC and ABCrf calibrate well marginally, while DNN is poorly calibrated in general
- Out of sample predictive performance is best for DNNC according to a certain “scoring rule”

Voles example - marginal calibration in simulated data



Future work

- Calibration for multivariate response
- Applications in likelihood-free inference:
  - Modelling dependence within components of the response using Gaussian copulas
  - Reducing dependence modelling to a sequence of univariate problems - likelihood-free Gibbs sampling (Rodrigues *et al.*, 2019) or by sequentially ordering components.
- Using the copula calibration with other more structured kinds of data (forecasting for time series, for example).

N. Klein, D.J. Nott and M.S. Smith. Marginally calibrated deep distributional regression, arXiv: 1908.09482, 2019.

Future work

- Calibration for multivariate response
- Applications in likelihood-free inference:
  - Modelling dependence within components of the response using Gaussian copulas
  - Reducing dependence modelling to a sequence of univariate problems - likelihood-free Gibbs sampling (Rodrigues *et al.*, 2019) or by sequentially ordering components.
- Using the copula calibration with other more structured kinds of data (forecasting for time series, for example).

N. Klein, D.J. Nott and M.S. Smith. Marginally calibrated deep distributional regression, arXiv: 1908.09482, 2019.

Future work

- Calibration for multivariate response
- Applications in likelihood-free inference:
  - Modelling dependence within components of the response using Gaussian copulas
  - Reducing dependence modelling to a sequence of univariate problems - likelihood-free Gibbs sampling (Rodrigues *et al.*, 2019) or by sequentially ordering components.
- Using the copula calibration with other more structured kinds of data (forecasting for time series, for example).

N. Klein, D.J. Nott and M.S. Smith. Marginally calibrated deep distributional regression, arXiv: 1908.09482, 2019.

Future work

- Calibration for multivariate response
- Applications in likelihood-free inference:
  - Modelling dependence within components of the response using Gaussian copulas
  - Reducing dependence modelling to a sequence of univariate problems - likelihood-free Gibbs sampling (Rodrigues *et al.*, 2019) or by sequentially ordering components.
- Using the copula calibration with other more structured kinds of data (forecasting for time series, for example).

N. Klein, D.J. Nott and M.S. Smith. Marginally calibrated deep distributional regression, arXiv: 1908.09482, 2019.

My research - likelihood-free inference

Bayesian statistical inference is "statistics as probability"

- Set up a full probability model for all data and unknowns
- Condition on the data after it's observed (Bayes rule)
- Use the resulting conditional distribution for the unknown after observing data (posterior density) for inference.

For many interesting models expressed generatively, the "likelihood" component in Bayes rule is intractable to compute.

- Can we use simulation from the model as a surrogate for not being able to compute the likelihood function?
- Yes, and there are various methods for doing so. Existing methods often don't work well in high dimensions.

My research - likelihood-free inference

Bayesian statistical inference is "statistics as probability"

- Set up a full probability model for all data and unknowns
- Condition on the data after it's observed (Bayes rule)
- Use the resulting conditional distribution for the unknown after observing data (posterior density) for inference.

For many interesting models expressed generatively, the "likelihood" component in Bayes rule is intractable to compute.

- Can we use simulation from the model as a surrogate for not being able to compute the likelihood function?
- Yes, and there are various methods for doing so. Existing methods often don't work well in high dimensions.

My research - likelihood-free inference

Bayesian statistical inference is "statistics as probability"

- Set up a full probability model for all data and unknowns
- Condition on the data after it's observed (Bayes rule)
- Use the resulting conditional distribution for the unknown after observing data (posterior density) for inference.

For many interesting models expressed generatively, the "likelihood" component in Bayes rule is intractable to compute.

- Can we use simulation from the model as a surrogate for not being able to compute the likelihood function?
- Yes, and there are various methods for doing so. Existing methods often don't work well in high dimensions.

My research - likelihood-free inference

Bayesian statistical inference is “statistics as probability”

- Set up a full probability model for all data and unknowns
- Condition on the data after it’s observed (Bayes rule)
- Use the resulting conditional distribution for the unknown after observing data (posterior density) for inference.

For many interesting models expressed generatively, the “likelihood” component in Bayes rule is intractable to compute.

- Can we use simulation from the model as a surrogate for not being able to compute the likelihood function?
- Yes, and there are various methods for doing so. Existing methods often don’t work well in high dimensions.

My research - likelihood-free inference

Bayesian statistical inference is "statistics as probability"

- Set up a full probability model for all data and unknowns
- Condition on the data after it's observed (Bayes rule)
- Use the resulting conditional distribution for the unknown after observing data (posterior density) for inference.

For many interesting models expressed generatively, the "likelihood" component in Bayes rule is intractable to compute.

- Can we use simulation from the model as a surrogate for not being able to compute the likelihood function?
- Yes, and there are various methods for doing so. Existing methods often don't work well in high dimensions.

My research - variational approximation

- Complex models for dependent data often involve observation specific latent variables (random effects models, state space models).
- Computations for these models is challenging, involving integrating over the space of the latent variables.
- Variational approximation methods compute posterior inferences by solving optimization problems.

Main challenge:

Define variational families with flexibility in ways relevant to the model at hand, but with parsimonious parametrizations leading to tractable optimization problems.

My research - variational approximation

- Complex models for dependent data often involve observation specific latent variables (random effects models, state space models).
- Computations for these models is challenging, involving integrating over the space of the latent variables.
- Variational approximation methods compute posterior inferences by solving optimization problems.

Main challenge:

Define variational families with flexibility in ways relevant to the model at hand, but with parsimonious parametrizations leading to tractable optimization problems.

My research - variational approximation

- Complex models for dependent data often involve observation specific latent variables (random effects models, state space models).
- Computations for these models is challenging, involving integrating over the space of the latent variables.
- Variational approximation methods compute posterior inferences by solving optimization problems.

Main challenge:

Define variational families with flexibility in ways relevant to the model at hand, but with parsimonious parametrizations leading to tractable optimization problems.

My research - variational approximation

- Complex models for dependent data often involve observation specific latent variables (random effects models, state space models).
- Computations for these models is challenging, involving integrating over the space of the latent variables.
- Variational approximation methods compute posterior inferences by solving optimization problems.

Main challenge:

Define variational families with flexibility in ways relevant to the model at hand, but with parsimonious parametrizations leading to tractable optimization problems.

Thank you

References

Z. An, L.F. South and C.C. Drovandi. BSL: Bayesian Synthetic Likelihood. R package version 2.0.0, 2019.

C. Bishop. Mixture density networks, Technical Report NCRG/4288, Aston University, Birmingham, UK, 1994

F. Chollet and J.J. Allaire. Deep Learning with R, 1st edn, Manning Publications Co., Greenwich, CT, USA, 2018.

K. Csillery, O. Francóis and M.G.B. Blum. ABC: an R package for approximate Bayesian computation (ABC), Methods in Ecology and Evolution 3: 475–479, 2012.

M. De Iorio, P. Ḿuller, G.L. Rosner and S.N. MacEachern. An ANOVA Model for Dependent Random Measures, Journal of the American Statistical Association, 99:205-215, 2004.

M. Fasilio and S. Wood. ABC in ecological modelling, in S. A. Sisson, Y. Fan and M. Beaumont (eds), Handbook of Approximate Bayesian Computation, Chapman and Hall, CRC, Boca Raton, 597–623, 2018.

References

N.J. Foti and S.A. Williamson. A survey of non-exchangeable priors for Bayesian nonparametric models, IEEE Transactions on Pattern Analysis and Machine Intelligence, 37:359–371, 2015.

T. Gneiting, F. Balabdaoui and A.E. Raftery. Probabilistic forecasts, calibration and sharpness, Journal of the Royal Statistical Society Series B, 69:243–268, 2007.

R. Jacobs, M.I. Jordan, S. Nowlan and G. Hinton. Adaptive Mixture of Local Expert. Neural Computation, 3:78-88, 1991

A. Kendall and Y. Gal. What uncertainties do we need in Bayesian deep learning for computer vision? Advances in Neural Information Processing Systems 31:5580–5590, 2017.

N. Klein, D.J. Nott and M.S. Smith. Marginally calibrated deep distributional regression, arXiv: 1908.09482, 2019.

References

R. Koenker and G. Bassett. Regression quantiles. Econometrica, 46:33–50, 1978.

V. Kuleshov, N. Fenner and S. Ermon. Accurate uncertainties for deep learning using calibrated regression, Proceedings of the 35th International Conference on Machine Learning, ICML, pp. 2801–2809, 2018

J.-M. Marin, L. Raynal, P. Pudlo, C.P. Robert and A. Estoup. abcrf: approximate Bayesian computation via random forests. R package version 1.7, 2017.

N. Meinshausen. Quantile regression forests, Journal of Machine Learning Research, 7:983–999, 2006.

A.H. Murphy and R.L. Winkler. A general framework for forecast verification. Monthly Weather Review, 115:1330-1338, 1987.

References

R.A. Rigby and D.M. Stasinopoulos. Generalized additive models for location, scale and shape, Journal of the Royal Statistical Society Series C, 54:507–554, 2005.

G.S. Rodrigues, D.J. Nott and S.A. Sisson. Likelihood-free approximate Gibbs sampling, arXiv: 1906.04347.

A. Sklar. Fonctions de repartition a n dimensions et leur marges, Publ. Inst. Statist. Univ. Paris 8, 229231 8: 229–231, 1959.

N. Tagasovska and D. Lopez-Paz. Frequentist uncertainty estimates for deep learning, arXiv:1811.00908, 2018.

P. Turchin and S.P. Ellne. Living on the edge of chaos: Population dynamics of fennoscandian voles, Ecology 81:3099–3116, 2000.