Marginaly calibrated deep distributional regression

Nadja Klein\textsuperscript{1}, David Nott\textsuperscript{2} and Michael Smith\textsuperscript{3}

\textsuperscript{1}Humboldt University of Berlin, \textsuperscript{2}National University of Singapore and \textsuperscript{3}University of Melbourne

February 18th, 2020
Parametric regression models

- Predict or explain variation in a response variable $Y$ based on features $X = (X_1, \ldots, X_p)$.
- We observe a training set of response and feature pairs $(y_i, x_i)$, $i = 1, \ldots, n$.
- A regression model estimates a conditional distribution of $Y$, given $X = x$, from the training data.
Typically only the mean response is modelled explicitly as a function of $x$.

“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]
Allow aspects of the distribution of $y$ beyond the mean to vary with $x$

- Complex parametric families where higher order moments are $x$-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 $p$. 
Allow aspects of the distribution of $y$ beyond the mean to vary with $x$

- Complex parametric families where higher order moments are $x$-dependent (e.g., 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 $p$. 
Allow aspects of the distribution of \( y \) beyond the mean to vary with \( x \):

- Complex parametric families where higher order moments are \( x \)-dependent (e.g. 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 $x$

- Complex parametric families where higher order moments are $x$-dependent (e.g., 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 $p$. 
Allow aspects of the distribution of $y$ beyond the mean to vary with $x$

- Complex parametric families where higher order moments are $x$-dependent (e.g., 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 $x$

- Complex parametric families where higher order moments are $x$-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 $x$

- Complex parametric families where higher order moments are $x$-dependent (e.g., 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 $p$. 
Deep neural network (DNN) - defines a flexible parametrized function $f_\eta(x)$ where $\eta$ denote some learnable parameters (weights).

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

For predicting scalar $Y$ from $x$, $f_\eta(x)$ will be a scalar-valued function.

$f_\eta(x)$ is a composition of component functions (“layers”): outputs from one layer feed into the next.
Deep neural network (DNN) - defines a flexible parametrized function $f_\eta(x)$ where $\eta$ denote some learnable parameters (weights).

For predicting scalar $Y$ from $x$, $f_\eta(x)$ will be a scalar-valued function.

$f_\eta(x)$ 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 $\eta$

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

- 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 $f_\eta(x)$ is the conditional mean response.
- For simple uncertainty quantification, a plug-in Gaussian predictive density is often used, after estimating the variance.
Minimize penalized empirical loss function with respect to $\eta$

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

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 $f_\eta(x)$ is the conditional mean response.

For simple uncertainty quantification, a plug-in Gaussian predictive density is often used, after estimating the variance.
Minimize penalized empirical loss function with respect to $\eta$

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

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 $f_\eta(x)$ 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 $\eta$

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

- 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 $f_\eta(x)$ 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 (e.g. 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 (e.g. 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 (e.g. 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.
Calibration means that forecasts should be statistically consistent with the observations. There are different kinds of calibration.

- Probability calibration - events given probability $p$ 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.
Calibration means that forecasts should be statistically consistent with the observations. There are different kinds of calibration.

- Probability calibration - events given probability $p$ 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 probability $p$ 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.
Calibration means that forecasts should be statistically consistent with the observations. There are different kinds of calibration.

- Probability calibration - events given probability $p$ 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.
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.
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.
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 \( Y = (Y_1, \ldots, Y_n) \) of a continuous response, with feature values \( x = \{x_1, \ldots, x_n\} \)

\[
p(y|x) = c^\dagger(F(y_1|x_1), \ldots, F(y_n|x_n)|x) \prod_{i=1}^{n} p(y_i|x_i)
\]

\( c^\dagger \) density of a parametric copula with parameters \( \theta \): we use an implicit copula of a deep neural network regression model, \( c_{\text{DNN}}(u|x, \theta) \)

Assume the distribution of \( Y_i|x_i \) has an invariant margin, so that \( p(y_i|x_i) = p_Y(y_i) \) with distribution function \( F_Y \)

\[
p(y|x, \theta) = c_{\text{DNN}}(F_Y(y_1), \ldots, F_Y(y_n)|x, \theta) \prod_{i=1}^{n} p_Y(y_i)
\]
Copula model

Consider observations $Y = (Y_1, \ldots, Y_n)^\top$ of a continuous response, with feature values $x = \{x_1, \ldots, x_n\}$

$$p(y|x) = c^\dagger(F(y_1|x_1), \ldots, F(y_n|x_n)|x) \prod_{i=1}^{n} p(y_i|x_i)$$

$c^\dagger$ density of a parametric copula with parameters $\theta$: we use an implicit copula of a deep neural network regression model, $c_{DNN}(u|x, \theta)$

Assume the distribution of $Y_i|x_i$ has an invariant margin, so that $p(y_i|x_i) = p_Y(y_i)$ with distribution function $F_Y$

$$p(y|x, \theta) = c_{DNN}(F_Y(y_1), \ldots, F_Y(y_n)|x, \theta) \prod_{i=1}^{n} p_Y(y_i)$$
Copula model

Consider observations $Y = (Y_1, \ldots, Y_n)^\top$ of a continuous response, with feature values $x = \{x_1, \ldots, x_n\}$

$$p(y|x) = c^\top(F(y_1|x_1), \ldots, F(y_n|x_n)|x) \prod_{i=1}^{n} p(y_i|x_i)$$

$c^\top$ density of a parametric copula with parameters $\theta$: we use an implicit copula of a deep neural network regression model, $c_{DNN}(u|x, \theta)$

Assume the distribution of $Y_i|x_i$ has an invariant margin, so that $p(y_i|x_i) = p_Y(y_i)$ with distribution function $F_Y$

$$p(y|x, \theta) = c_{DNN}(F_Y(y_1), \ldots, F_Y(y_n)|x, \theta) \prod_{i=1}^{n} p_Y(y_i)$$
Copula model

Consider observations $Y = (Y_1, \ldots, Y_n)^\top$ of a continuous response, with feature values $x = \{x_1, \ldots, x_n\}$

$$p(y|x) = c^\dagger(F(y_1|x_1), \ldots, F(y_n|x_n)|x) \prod_{i=1}^{n} p(y_i|x_i)$$

$c^\dagger$ density of a parametric copula with parameters $\theta$: we use an implicit copula of a deep neural network regression model, $c_{\text{DNN}}(u|x, \theta)$

Assume the distribution of $Y_i|x_i$ has an invariant margin, so that $p(y_i|x_i) = p_Y(y_i)$ with distribution function $F_Y$

$$p(y|x, \theta) = c_{\text{DNN}}(F_Y(y_1), \ldots, F_Y(y_n)|x, \theta) \prod_{i=1}^{n} p_Y(y_i)$$
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

(a) Probability calibration

(b) Marginal calibration

David Nott, NUS
Boston housing data
Predictive densities

(a) DNN
(b) DNN recalibrated
(c) DNCC ridge
(d) DNCC horseshoe
Let $\rho$ denote the parameters in a parametric statistical model for data $d$ with density $p(d|\rho)$.

**Bayesian inference**

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

- Full probability model: $p(\rho, d) = p(\rho)p(d|\rho)$, where $p(\rho)$ is a prior density expressing what is known before observing data.
- For inference, use the *posterior density* $p(\rho|d_{\text{obs}})$. 
Let $\rho$ denote the parameters in a parametric statistical model for data $d$ with density $p(d|\rho)$.

Bayesian inference

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

- Full probability model: $p(\rho, d) = p(\rho)p(d|\rho)$, where $p(\rho)$ is a prior density expressing what is known before observing data.
- For inference, use the posterior density $p(\rho|d_{\text{obs}})$. 
Let $\rho$ denote the parameters in a parametric statistical model for data $d$ with density $p(d|\rho)$

**Bayesian inference**

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

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

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

Bayesian inference

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

- Full probability model: $p(\rho, d) = p(\rho)p(d|\rho)$, where $p(\rho)$ is a prior density expressing what is known before observing data.
- For inference, use the posterior density $p(\rho|d_{\text{obs}})$. 
Frequently, interesting models are specified in terms of how you generate data from them rather through an explicit form for $p(d|\rho)$.

**Likelihood-free Bayesian computation by regression**

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

- We don’t need to compute $p(d|\rho)$ anywhere.
- We use our regression copula approach for the distributional regression. Motivation: the true posterior is marginally calibrated.
Frequently, interesting models are specified in terms of how you generate data from them rather through an explicit form for $p(d|\rho)$.

**Likelihood-free Bayesian computation by regression**

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

- We don’t need to compute $p(d|\rho)$ anywhere.
- We use our regression copula approach for the distributional regression. Motivation: the true posterior is marginally calibrated.
Frequently, interesting models are specified in terms of how you generate data from them rather through an explicit form for $p(d|\rho)$.

**Likelihood-free Bayesian computation by regression**

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

We don’t need to compute $p(d|\rho)$ 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 $p(d|\rho)$.

**Likelihood-free Bayesian computation by regression**

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

- We don’t need to compute $p(d|\rho)$ anywhere.
- We use our regression copula approach for the distributional regression. Motivation: the true posterior is marginally calibrated.
Trapping data on Voles abundance from Kilpisjarvi, Finland.

90 data points collected during spring and autumn of each year, 1952-1997

![Graph showing trapping index from 1960 to 1990](image)

Figure 3: Top: observed voles trapping index in Kilpisjarvi, between 1952 and 1997. Middle and bottom: two realizations (solid and dashed) of model 6, using parameters equal to the posterior means given by SLMH and PMMH.

Season, divided by the number of hundred trap-nights used in that season. After 1980 the number of trap-nights was fixed to around 1000, but in earlier years this number is not available: it varied from a minimum of 500 to more than a thousand (Perry, 2000). This correction for the sampling effort implies that, if the number of the trapped voles in each season is approximately Poisson distributed, the trapping index is not.

We have dealt with this problem by multiplying the data in Figure 3 by 10 and rounding each data-point to the nearest integer. This solution should give near-exact results for data collected after 1980, and a good approximation for all data-points representing a considerable population, thanks to the normal approximation to the Poisson distribution.

A useful source of prior information is represented by Turchin and Hanski (1997), where life history and data from short experiments were used to estimate the parameters of model (7). We report the prior distributions for each parameter in Table 1. The expected values of the prior distributions have been chosen on the basis of the remarks of Turchin and Hanski (1997), and we refer the reader to this reference for further details.

The specific distributions and variabilities used for the priors have been chosen based on an attempt at quantifying the remarks of Turchin and Hanski (1997) regarding their confidence in their independently derived estimates. Admittedly, this process entails a certain degree of arbitrariness. No prior information was available for \( \phi \) and \( \sigma \), hence we have used improper uniform priors for both parameters.

For SL we used the following set of 17 summary statistics:

- autocovariances of \( n_1, \ldots, n_T \) up to lag 5;
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
- \(a, \delta\) - maximal predation rate for individual weasels, half saturation prey density
- \(\phi\) sampling rate parameter, \(d_t \sim \text{Poisson}(\phi n_t)\).
- \(\sigma\) - standard deviation of driving Brownian motion
We use an implicit copula of a convolutional network to predict the components of the parameter vector \( \rho \in \mathbb{R}^p \) based on data \( d = (d_1^T, \ldots, d_T^T)^T \).

- Can be estimated based on training sets of simulations of pairs \((\rho_k, d_k)\) 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 \( \rho \in \mathbb{R}^p \) based on data \( d = (d_1^T, \ldots, d_T^T)^T \).

- Can be estimated based on training sets of simulations of pairs \((\rho_k, d_k)\) 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.
We use an implicit copula of a convolutional network to predict the components of the parameter vector $\rho \in \mathbb{R}^p$ based on data $d = (d_1^T, \ldots, d_T^T)^T$.

Can be estimated based on training sets of simulations of pairs $(\rho_k, d_k)$ 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 $\rho \in \mathbb{R}^p$ based on data $d = (d_1^T \ldots d_T^T)^T$
- Can be estimated based on training sets of simulations of pairs $(\rho_k, d_k)$ 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

(a) $r$

(b) $e$

(c) $g$

(d) $h$

(e) $a$

(f) $\delta$

(g) $s$

(h) $\sigma$

(i) $\psi$
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).

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).

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).

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).

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.
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.
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.
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.
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.
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.
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.
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.
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


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


