The Role of Statistics in the Growing Realm of Data Science

Jürgen Pilz

Institut für Statistik, Universität Klagenfurt
Universitätsstr. 65-67, 9020 Klagenfurt, Austria
juergen.pilz@aau.at

Int. Conference on Recent Trends in Statistics and Data Analytics
Sept 22-23, 2022 / Islamabad, Pakistan

I. Introduction

Over the last decade, many conferences under the heading of "Machine Learning/Deep Learning", "Data Science", "Data Analysis/Analytics", "Big Data", but much fewer with joint theme "**Statistics** and Data Science", "**Statistics** and Data Analytics", and still fewer with sole theme "(Applied) Statistics".

Good message: Joint theme conferences (research monographs, textbooks, ...) and university curricula of Data Science study programs incl. sound education in statistics are gaining ground!

Main Drivers of recent developments/ trends have been

- Availability of massive data sets
- Modern Computer Technology and Computing Environments

Essential role of Probability Theory, Information Theory and Statistics is getting more and more acknowledged! But, we need to double our efforts to propagate the underlying probabilistic and statistical basis of Data Science and and our contributions to it!

Introduction

"Statistics is the grammar of science." Karl Pearson (1892)

"Those who ignore statistics are condemned to reinvent it. Statistics is the science of learning from experience." Bradley Efron (2006)

"Data can tell lies. – Big Data can tell bigger lies. – The big thing for small data is random error. – The big thing for big data is bias." Chris Wild (2017)

Fundamental ideas in statistics: uncertainty and variation.
Two of these key developments over the last decades are **bootstrapping** (Bradley Efron, 1979) and **Monte Carlo Markov Chain** (MCMC, Gelfand and Smith 1990) methods, which make it possible to compute large hierarchical models, e.g. in Bayesian statistics, computational physics and chemistry, computational biology and linguistics, etc.

The widespread use of such powerful computational tools would have been impossible without the emergence of the statistical programming language R (released in 1993)

Introduction

Creative Task of Statisticians: Data **Modelling**
Note: "All models are wrong, but some of them are useful"
(George E.P. Box 1978)

Looking into Data Science/ML books: Regression and Classification (Models) dominate the contents

Basically, the underlying concept for both is the same:

Conditional Expectation $\mathbb{E}[Y|x_1,\ldots,x_k] = f(x_1,\ldots,x_k)$

Continuous Y: Regression case
Discrete (multinomial) Y: Classification case

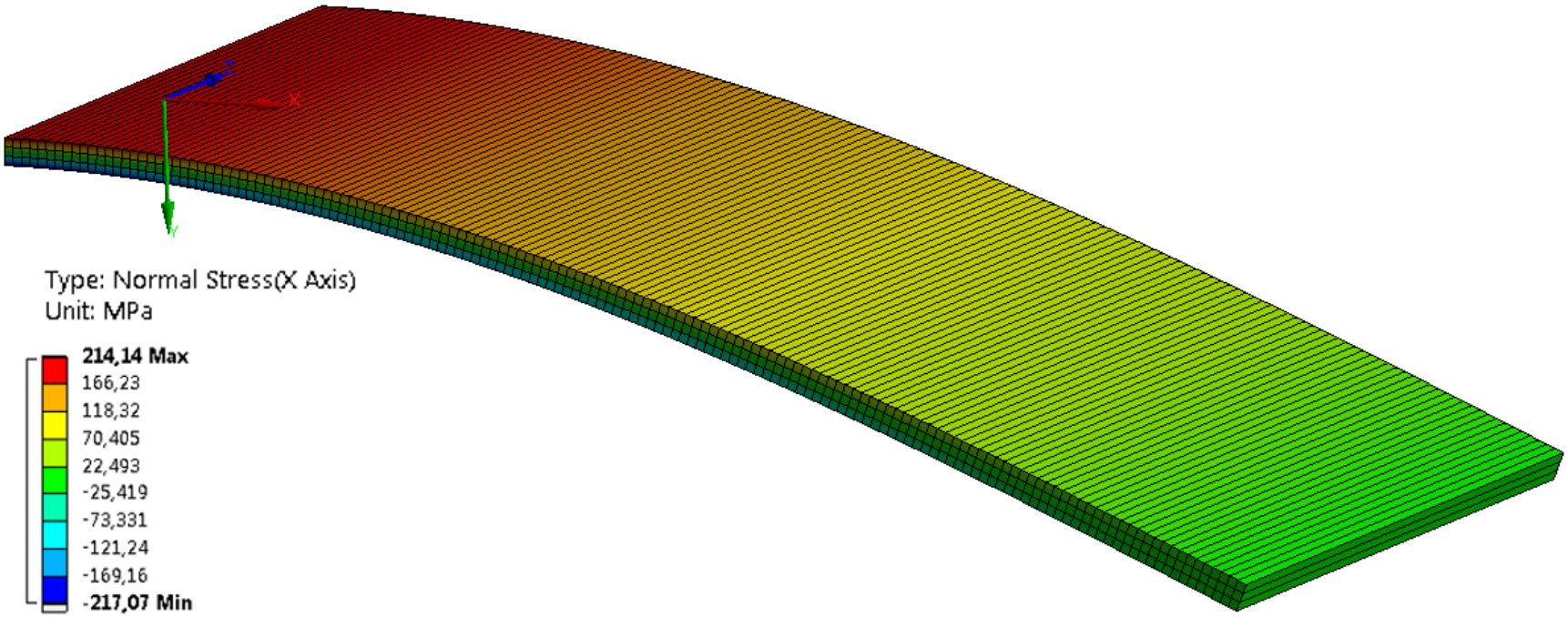
In this talk: deal with both central topics

Regression: Linear regression ... Gen. linear (mixed) regression
... Additive regression ... Gaussian Process regression

Classification: Clustering ... Bayes Deep Learning

II. Gaussian Process Regression

Start with specific application:
Stress testing in semiconductor processing for **thin wafers** (thickness$\leq 40\mu m$)
Kriging metamodel for stress prediction validated against Ramann spectroscopy measurements, FEM simulations
+ Modelling of electrical parameters (signals)



Gaussian Process Regression

- **Experiments**
  - physical experiments
  - Computer experiments (Computer-based simulations like FEM)

Which simulations to run?
Main difference: Computer Models are **deterministic**
Modification of classical DOE $\Rightarrow$ DOCE

Math. model: $y = \tilde{f}(x_1 \ldots, x_k)$, e.g. solution of ODE/PDE system

$$\mathbf{x} = (x_1, \ldots, x_k)^T \in \mathcal{X} = \text{experimental domain}$$

replaced by *meta-model*

$$\mathbb{E}\, Y(\mathbf{x}) = f(x_1, \ldots, x_k), f \text{ "close" to } \tilde{f}$$

Gaussian Process Regression

Requirements for good designs:

- space filling property
- projective property
- computational efficiency

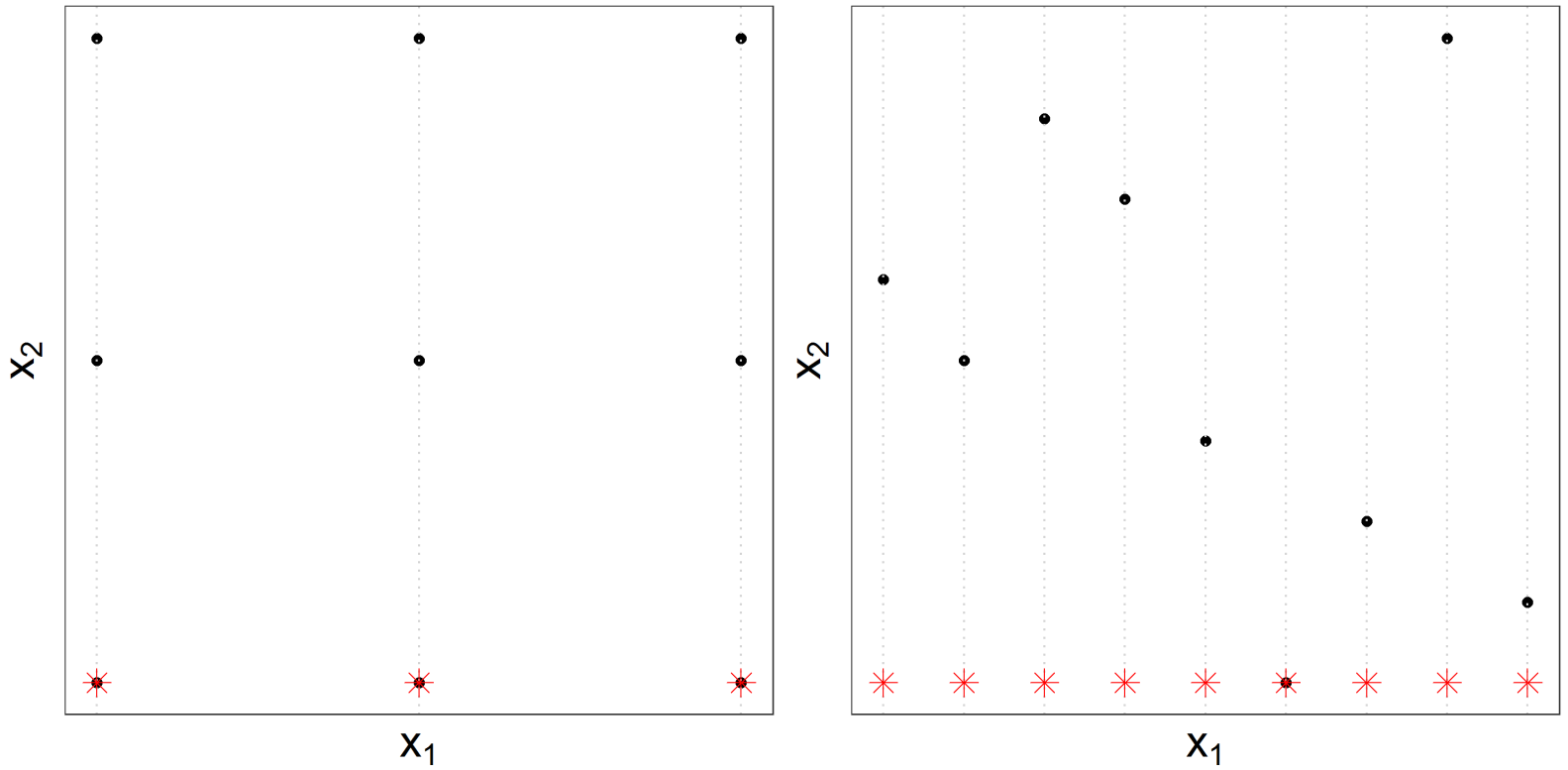
Compromise: **LHD**= Latin Hypercube Designs

w.l.o.g. experimental domain $\mathcal{X} = [0, 1]^k$

Designs: $d_n = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathcal{X}^n$,

n runs, k factors

Gaussian Process Regression



Regular (left) and latin hypercube design (right)

Gaussian Process Regression

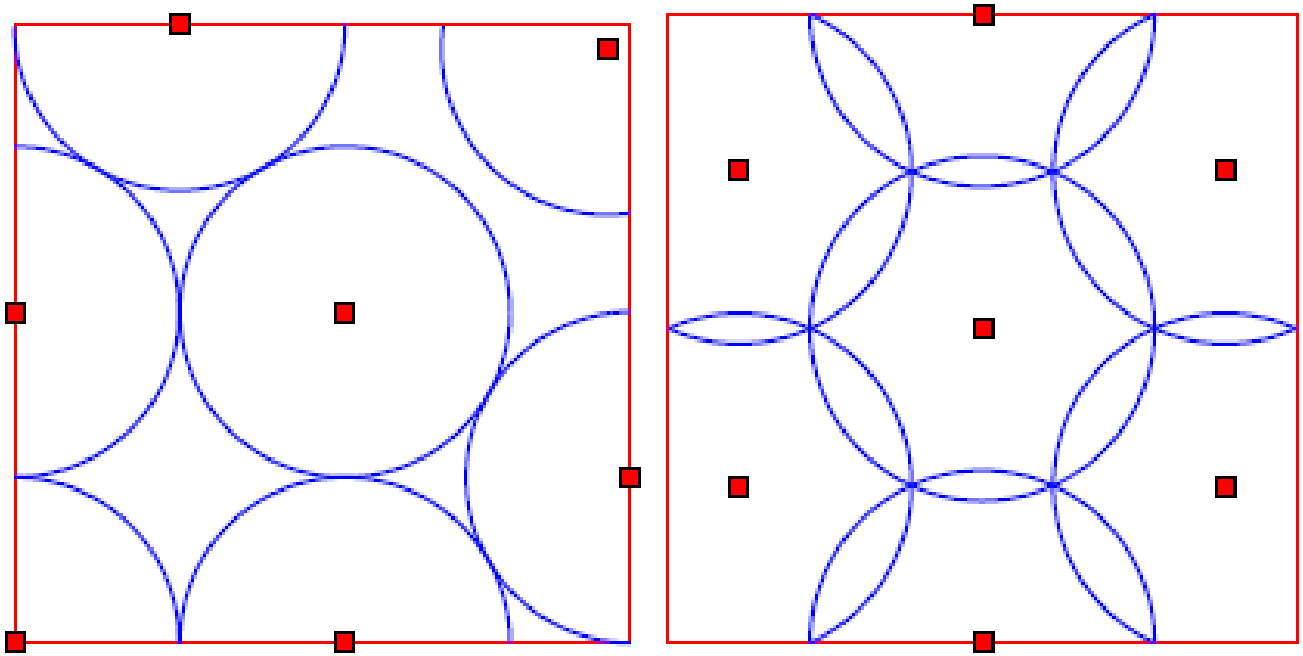
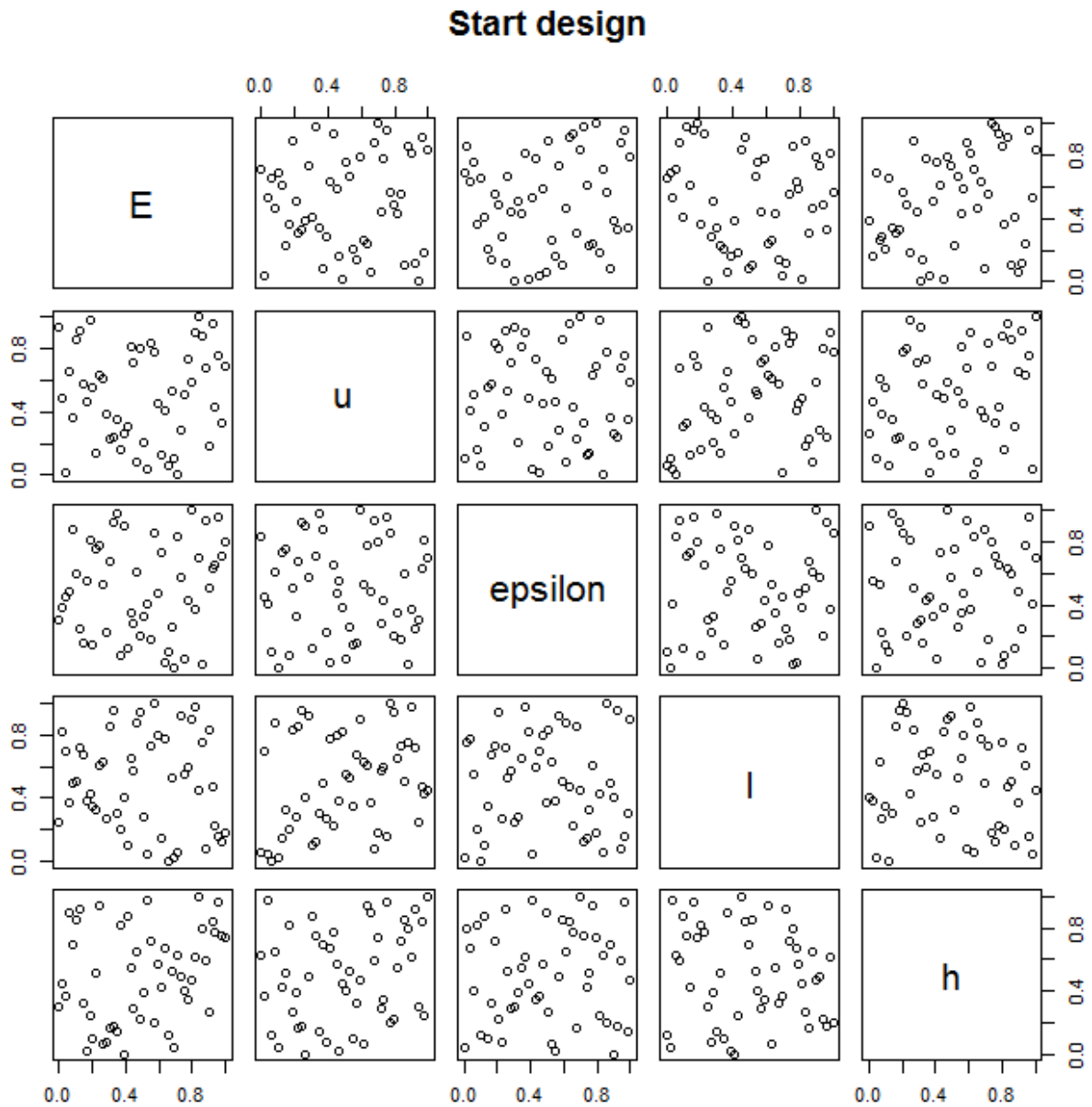
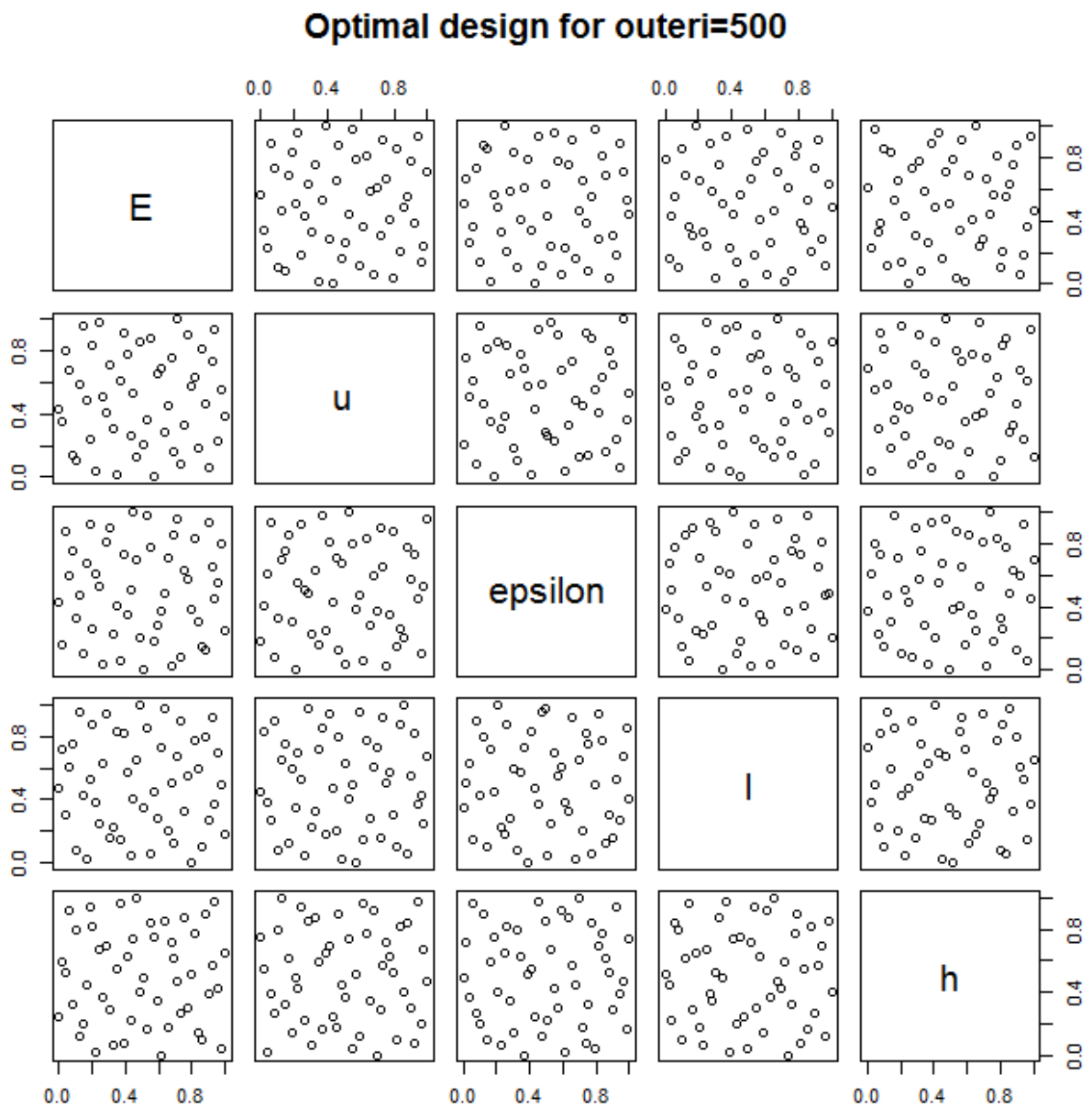


Fig.. Maximin (left) and Minimax (right) designs

Gaussian Process Regression



Gaussian Process Regression



RSM modelling

Classical approach: Regression (response surface) modelling $\Rightarrow$

prediction reduces to interpolation problem

e.g. quadratic RSM

$$y(\mathbf{x}) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i<j}^{k} \sum_{j=1}^{k} \beta_{ij} x_i x_j$$

For complex responses, LSE $\hat{\beta} = (X^T X)^{-1} X^T \mathbf{y}_d$

yields bad interpolations

Remedy

Kriging models

$\Rightarrow$ response = realization of stochastic process

$$Y(\mathbf{x}) = \underbrace{\mu(\mathbf{x})}_{\text{trend}} + \underbrace{Z(\mathbf{x})}_{\substack{\text{Gaussian Process} \\ \text{(zero mean)}}}$$

Effect: good approx. over a wide range of different designs and sample sizes and well-defined basis for statistical framework

$$Y(\cdot) \sim GP(\mu(\mathbf{x}), \sigma^2 R(\cdot, \cdot))$$

Main difference to geostatistical settings:
– $\mathbf{x}$ is no spatial coordinate vector
– usually, higher dimensional settings: $K > 3$

Covariance structure

covariance function: $Cov(Z(\mathbf{x}_i), Z(\mathbf{x}_j)) = \sigma^2 \underbrace{R(\mathbf{x}_i, \mathbf{x}_j)}_{\text{correlation function}}$

$\downarrow$

process variance

Common assumptions:

1. covariance-stationarity, i.e.

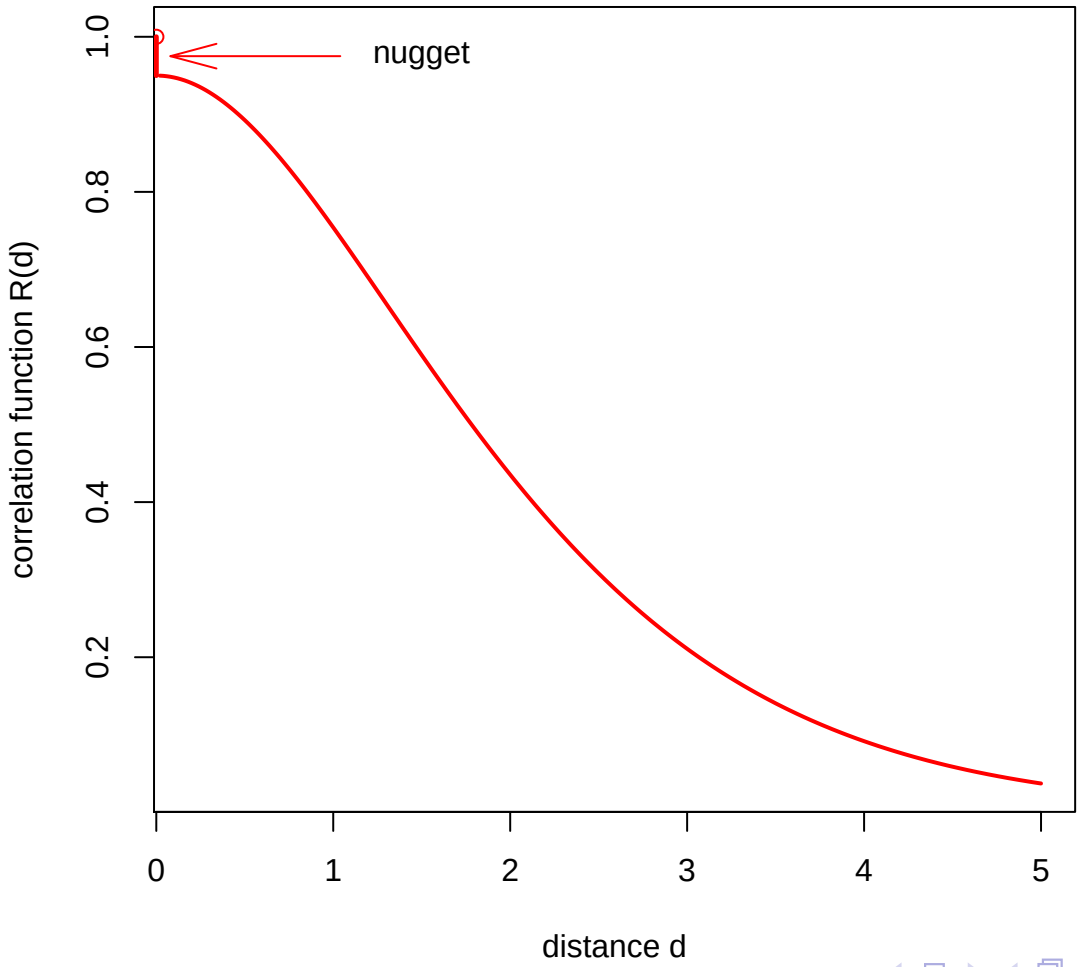
$$R(\mathbf{x}_i, \mathbf{x}_j) = R(\mathbf{x}_i - \mathbf{x}_j)$$

2. (tensor-)product correlation structure

$$R(\mathbf{x}_i, \mathbf{x}_j) = \prod_{m=1}^{k} \underbrace{R_m(|x_{im} - x_{jm}|)}_{\text{univariate Matérn c.f.}}$$

Matérn c.f. $\nu = \frac{5}{2}$

$$R(d) = \left(1 - \frac{\tau^2}{\sigma^2}\right) * \left(1 + \frac{\sqrt{5}d}{\theta} + \frac{5d^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}d}{\theta}\right), \; d > 0$$



Estimation and Prediction

MLE: available for β and σ^2

$\hat{\boldsymbol{\beta}} = (X^T R_n^{-1}(\theta) X)^{-1} X^T R_n^{-1}(\theta) \mathbf{y}_d$

$\hat{\sigma}^2 = \frac{1}{n}(\mathbf{y}_d - X\hat{\boldsymbol{\beta}})^T R_n^{-1}(\theta)(\mathbf{y}_d - X\hat{\boldsymbol{\beta}})$

$\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_k)^T$ Gauss-Newton (or genetic optimiz.)

Optimal prediction:

$$\begin{aligned} \hat{Y}(\mathbf{x}^*) &= \mathbf{f}(x^*)^T \hat{\boldsymbol{\beta}} + \mathbf{r}_0^T R_n^{-1}(\mathbf{y}_d - X\hat{\boldsymbol{\beta}}) \\ &= \text{GLSE + smoothed residual} \end{aligned}$$

where $\mathbf{r}_0^T = (R(\mathbf{x}_0 - \mathbf{x}_1), \ldots, R(\mathbf{x}_0 - \mathbf{x}_n))$, $R_n =$ correl. matrix

Implementation in R: **DiceKriging**

Robust Additive Gaussian Processes

Aims

- higher flexibility in meta-modelling
- numerical stability: robustness of parameter estimates, esp. for correlation parameters

Solution: Bayesian approach using additive models and (objective) reference priors

Side effect: high-dimensional optimization problems reduced to a few sub-routines of ≤ 3 dimensions

Additive model:

$$\mathbb{E}\,Y(\mathbf{x}) = f_0 + \sum_{i=1}^{k} f_i(x_i) + \sum_{i<j} f_{ij}(x_i, x_j) + \ldots + f_{12\ldots k}(x_{1,\ldots}, x_k)$$

Functional ANOVA Representation

Robust Additive Gaussian Processes

Special case: first order GAM

$$\mathbb{E}\, Y(x_1, \ldots, x_k) = f_0 + \sum_{i=1}^{k} f_i(x_i)$$

$f_1, ..., f_k$: smooth basis functions

$\Rightarrow$ non-parametric modelling of main effects

Goal: Extension of classical GAM regression

Fo a good overview of the advantages of additive structures compared to fully parametric GP models in high dimensions see Dourante, Ginsbourger, Roustant (2012)

Robust Additive Gaussian Processes

Novelty of our recently proposed concept: Combination of AGP with robust reference priors proposed by Gu, Wang and Berger (AS 2018) + new sampling design scheme

Our new model: Second order Kriging AGP with

$$f_i \sim N(\mu_i, \sigma^2 R_i)$$

$$f_{ij} \sim N(\mu_{ij}, \sigma^2 R_i R_j)$$

Result: AGP $Y(\mathbf{x}) \sim N(\mu, \sigma^2 R(\cdot,\cdot))$, locally constant trend

and $R(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{k} R_i(x_i, x_i') + \sum_{i=1}^{k} \sum_{j=i+1}^{k} R_i(x_i, x_i') R_j(x_j, x_j') + \delta_{\mathbf{x}\mathbf{x}'} \tau^2$

Robust Additive Gaussian Processes

Profile likelihood approach often fails!

Remedy: robust Bayes prediction using reference priors of the form

$$\pi^R(\mu, \sigma^2, \theta^*) = \frac{\pi^R(\theta^*)}{\sigma^2}$$

$\downarrow$

correl. parameters

where $\pi^R(\theta^*) \propto (det \;\; I_F(\theta^*))^{1/2}$

$\downarrow$

exp. Fisher information

Explicit representations for $I_F(\theta^*)$ available in Kazianka & Pilz (2012)

Result: proper posteriors $p(\theta^*|\mathbf{y}_d)$

Simplified estimate: $\hat{\theta}^* = arg \;\; \max_{\theta^*} \;\; p(\theta^*|\mathbf{y}_d)$

posterior mode (to avoid MCMC)

Bayes Prediction

Bayes predictor of $Y(\mathbf{x}_0)$ for untried input $\mathbf{x}_0$ is based on the predictive distribution

$$p(Y_0|\mathbf{y}_d) = \int \underbrace{p(Y_0|\mathbf{y}_d, \theta^*)}_{Student-t} p(\theta^*|\mathbf{y}_d) d\theta^*$$

R-implementation fully described in
Vollert, Ortner & Pilz (2019): Robust Additive Gaussian Process Models Using Reference Priors and Cut-Off-Designs, J. Applied Mathematical Modelling 65 (2019), 586-596

As a test function we used, among others:
Friedman function: $x_i \in [0,1]; i = 1, \ldots, 5$

$$f_3(\mathbf{x}) = 10 \sin(x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5$$

$n_3 = 47$ samples

Results

1. Cut-FD can better determine the actual structure of the test functions than maximin and other LHDs (found exact set of components for Pepelyshev and Friedman functions, maximin did not)
2. Maximin LHD design was best with regard to MAPE (pred. power): For 5D-Friedman function MAPE $< 4\%$ (based on only $n_3 = 47$ sample points!)
3. **Robust AGP** model **outperforms** commonly used **GP** models for all three test functions
4. Simple random LHDs are least appropriate for approximation, getting even worse with increasing dimension

Results

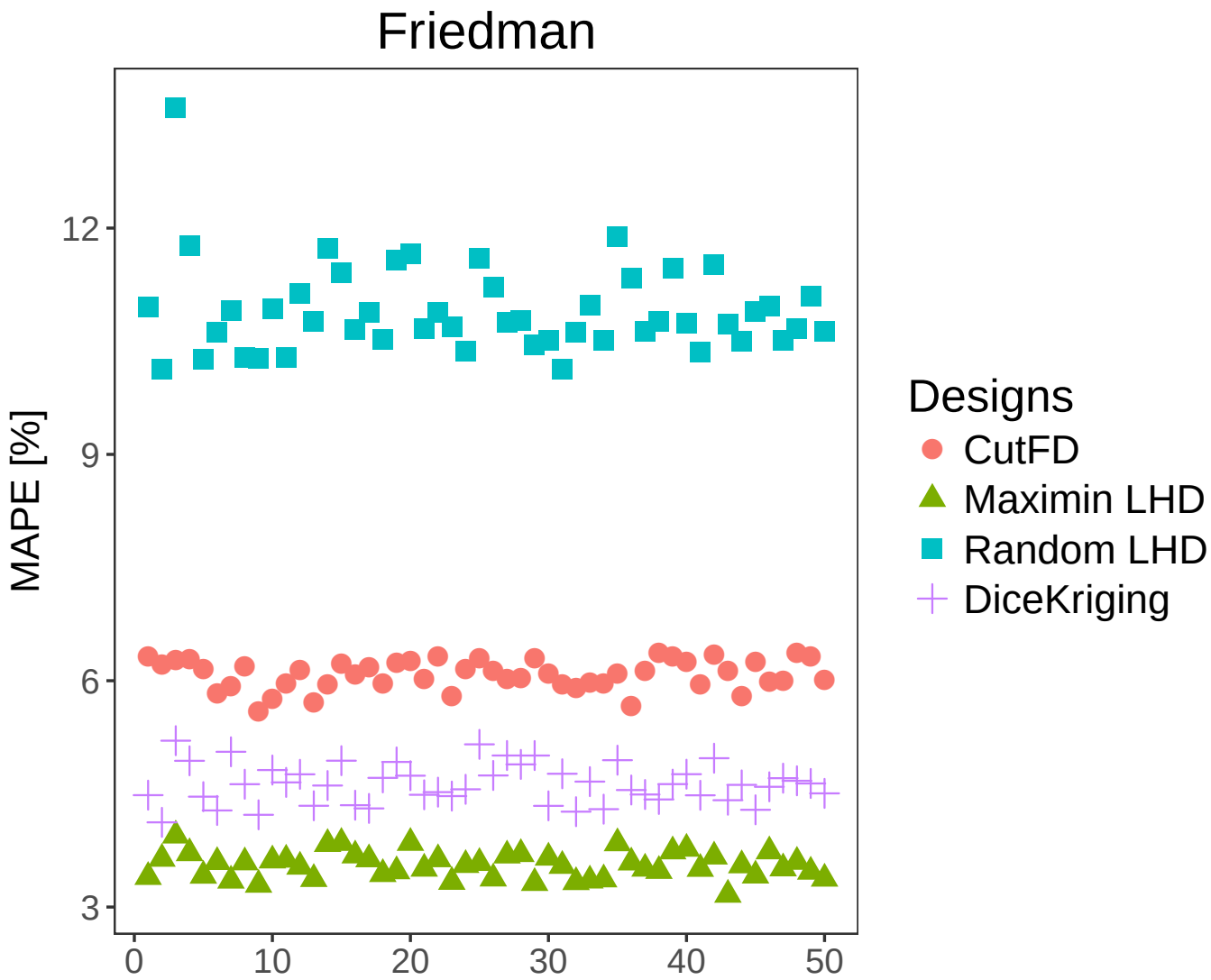
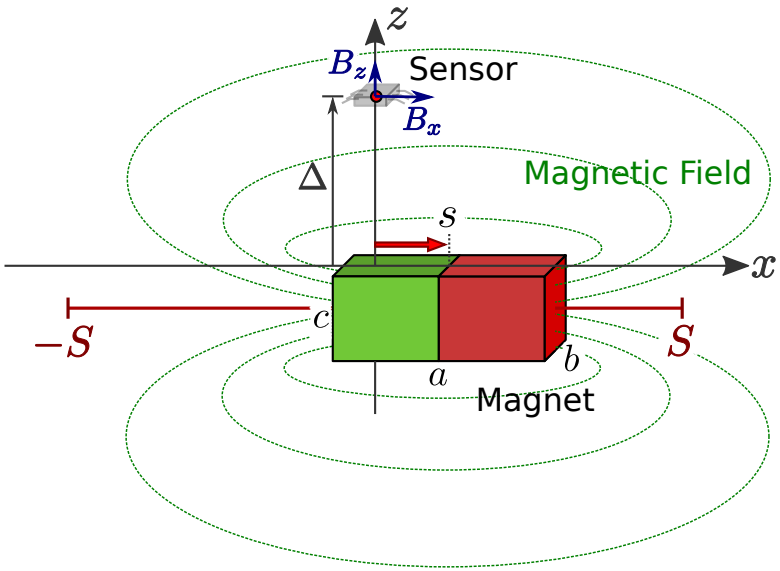


Fig.. MAPE values of 50 validation LHDs for 5D-Friedman test function

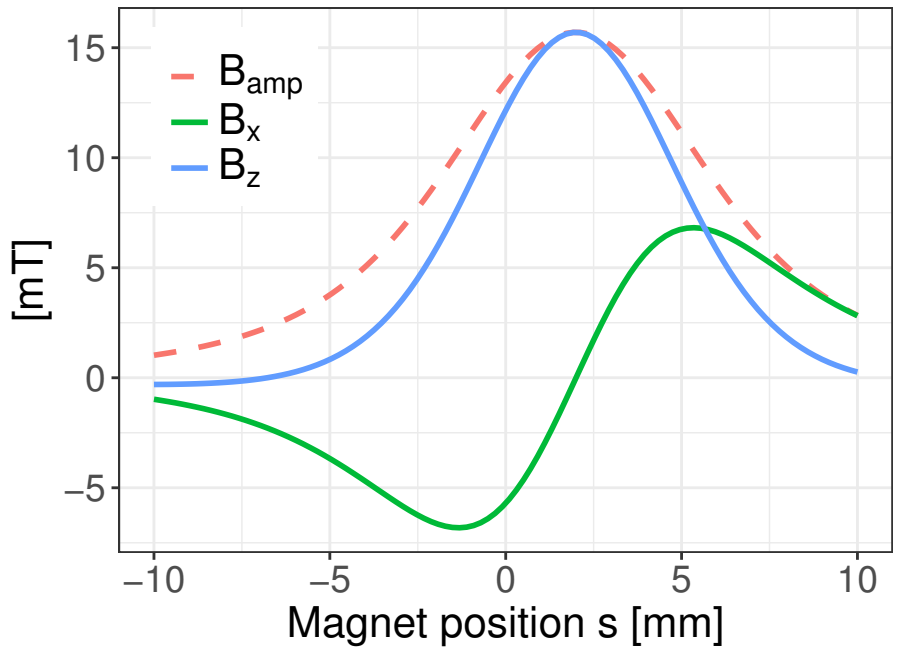
Automotive Application

AGP modelling for real DOCE applications based on FEMs for geometric and material parameter optimization problems, e.g. Magnetic field shaping for position and orientation detection systems

a)



b)



III. Multiple Linear Regression

Still the most widely used Regression model type
Focus here: variable selection for models with large number of regressors

Posch, Arbeiter and Pilz: A novel Bayesian approach for variable selection in linear regression models. CSDA 144 (2020)

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

Among the well-known recent approaches to variable selection the most popular one is **lasso**, proposed by Tibshirani (1996), with penalty $\mathcal{P}(\boldsymbol{\beta}, \lambda) := \lambda||\boldsymbol{\beta}||_1 = \lambda \sum_{i=1}^{p} |\beta_i|$.

Shrinkage effect!

The lasso can be viewed as a convex, more efficiently solvable reformulation of the best subset selection approach with penalty

$\mathcal{P}(\boldsymbol{\beta}, \lambda) := \lambda||\boldsymbol{\beta}||_0 = \lambda\#(i|\beta_i \neq 0)$

Bayesian lasso

lasso estimate for β can be interpreted as a Bayesian posterior mode estimate when independent Laplace priors all with zero mean and the same scale parameter $\lambda > 0$ are assigned to the coefficients:

$$p(\beta|\sigma^2) = \prod_{i=1}^{p} \frac{\lambda}{2\sigma} e^{-\lambda \frac{|\beta_i|}{\sigma}}$$

A further generalization is the adaptive lasso (Zou2006), which allows for different penalization factors of the regression coefficients:

$$\mathcal{P}(\beta, \boldsymbol{\lambda}) := \sum_{i=1}^{p} \lambda_i |\beta_i|.$$

The Bayesian adaptive lasso generalizes the non-adaptive Bayesian lasso by allowing different scale parameters in the Laplace priors:

$$p(\beta|\sigma^2) = \prod_{i=1}^{p} \frac{\lambda_i}{2\sigma} e^{-\lambda_i \frac{|\beta_i|}{\sigma}}.$$

Shrinkage control

Another generalization of classical lasso: elastic net, Hastie (2005), with penalty function given by:

$$\mathcal{P}(\boldsymbol{\beta}, \boldsymbol{\lambda}) := \lambda_1 ||\boldsymbol{\beta}||_1 + \lambda_2 ||\boldsymbol{\beta}||_2^2.$$

This encourages a grouping of strongly correlated predictors. It works better than the classical lasso when $p \gg n$. Bayesian versions of the elastic net include Huang (2015). We used their R-package *EBglmnet* to validate our novel method.
Also: Bayesian penalized regression techniques not directly related to the lasso e.g. the **horseshoe** estimator, see Makalic (2016)

$$\beta_j | \lambda_j, \tau, \sigma^2 \sim \mathcal{N}(0, \lambda_j^2 \tau^2 \sigma^2),$$
$$\lambda_j \sim \mathrm{C}^+(0, 1),$$
$$\tau \sim \mathrm{C}^+(0, 1)$$

τ controls amount of overall shrinkage of $\boldsymbol{\beta}$, while $\lambda_1, ..., \lambda_p$ allow for individual adaptions on the degree of shrinkage.

Random indicator approach

Zhang 2018 proposed Dirichlet-Laplace (DL) shrinkage priors, leading to optimal posterior concentration:

$$\begin{aligned} \beta_j|\phi_j, \tau, \sigma^2 &\sim \text{La}(\phi_j \tau \sigma), \\ (\phi_1, ..., \phi_p)^T &\sim \text{Dir}(a, ..., a), \\ \tau &\sim \text{Ga}\left(pa, \frac{1}{2}\right), \end{aligned}$$

Small values of concentration parameter a guarantee that only some of the components of $\phi = (\phi_1, ..., \phi_p)^T$ are nonzero.

Besides: Bayesian methods using a random indicator vector $\gamma = (\gamma_1, ..., \gamma_p)^T \in \{0, 1\}^p$ gain increasing popularity, see Wang (2015).

$\gamma_i = 0 \Rightarrow i$-th predictor does not explain the target y.

Common choice: independent Bernoulli priors for indicator variables:

Spike and slab

These Bayesian methods belong to the so-called **spike and slab** approaches: use mixture priors, with a spike concentrated around zero and a comparably flat slab, to perform variable selection.

Note: spike and slab priors are also applied apart from classical regression approaches. E.g. Polson (2011) used this type of priors to regularize support vector machines.

In our recent paper Posch, Arbeiter and Pilz (2020): setting is based on a random set $\mathcal{A} \subseteq \{1, ..., p\}$ that holds the indices of the active predictors, i.e. the predictors with coefficients different from zero. We assign a prior to $\mathcal{A}$ which depends on the cardinality of the set $|\mathcal{A}|$ as well as on the actual elements of $\mathcal{A}$:

$$p(\mathcal{A} = \{\alpha_1, ..., \alpha_k\}) \propto (p_{\alpha_1} + ... + p_{\alpha_k}) \frac{1}{k} \tilde{p}(k)$$

where $\tilde{p}$ is our a priori belief in the model size and

$\{p_{\alpha_1}, ..., p_{\alpha_k}\} \subseteq \{p_1, ..., p_p\}$ with $\sum_{i=1}^{p} p_i = 1$ and $p_i \geq 0$ for $i = 1, ..., p$

Zellner shrinkage

Special case: equal a priori importance of the predictors $\frac{1}{p} = p_1 = ... = p_p \Rightarrow$ prior reduces to

$$p(\mathcal{A} = \{\alpha_1, ..., \alpha_k\}) \propto \tilde{p}(k)$$

For the variance σ^2 of the error terms $\varepsilon_1, ..., \varepsilon_n$ an inverse gamma prior is chosen:

$$p(\sigma^2) \propto (\sigma^2)^{-(a+1)} \exp\left(-\frac{b}{\sigma^2}\right).$$

For given $\mathcal{A}$ the vector of nonzero coefficients $\beta_{\mathcal{A}}$ is commonly assigned a conventional Zellner g-prior

$$\beta_{\mathcal{A}} | g, \sigma^2, \mathbf{X}_{\mathcal{A}} \sim \mathcal{N}(\mathbf{0}, g\sigma^2(\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}})^{-1})$$

where $\mathbf{X}_{\mathcal{A}}$ = submatrix of $\mathbf{X}$ consisting of all columns corresponding to predictors with index in $\mathcal{A}$.

Penalized Zellner g-prior

However, to overcome problems with singularity of $\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}}$ for $k > n$, we consider a ridge penalized version of the g-prior:

$$\beta_{\mathcal{A}}|g, \sigma^2, \mathbf{X}_{\mathcal{A}} \sim \mathcal{N}(\mathbf{0}, (g^{-1}\sigma^{-2}\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}} + \lambda \mathrm{I}_k)^{-1}),$$

with small $\lambda > 0$ and complete hierarchical representation

$$p(\mathcal{A} = \{\alpha_1, ..., \alpha_k\}) \propto (p_{\alpha_1} + ... + p_{\alpha_k})\frac{1}{k}\tilde{p}(k),$$

$$g \sim \mathrm{IG}\left(\frac{1}{2}, \frac{n}{2}\right),$$

$$\sigma^2 \sim \mathrm{IG}\,(a, b)$$

Our main result: above model specifications are **consistent** in terms of model selection:

$$\operatorname*{p\,lim}_{n\to\infty} p(M_{\mathcal{A}}|\mathbf{y}, \mathbf{X}) = 1 \quad \text{and} \quad \operatorname*{p\,lim}_{n\to\infty} p(M_{\mathcal{A}'}|\mathbf{y}, \mathbf{X}) = 0 \quad \text{for all } \mathcal{A}' \neq \mathcal{A},$$

i.e. the true model will be selected provided that enough data is available.

MH algorithm

Implementation: special Metropolis-Hastings (MH) algorithm is proposed to simulate from

$p(\beta, \mathcal{A}, g, \sigma^2|\mathbf{y}, \mathbf{X})$.

Have to define transitions
$q(\alpha|\mathcal{A}_t), q(\mathcal{A}_{t+1}|\mathcal{A}_t, c_h = 1), q(\mathcal{A}_{t+1}|\mathcal{A}_t, c_h = 0) \Rightarrow q(\mathcal{A}_{t+1}|\mathcal{A}_t)$.

Natural choice for proposal distribution of $\beta_{t+1}|\mathcal{A}_{t+1}, g_{t+1}, \sigma^2_{t+1}, \mathbf{X}_{\mathcal{A}_{t+1}}$ is normal with mean and precision matrix given by

$$\boldsymbol{\mu}_{t+1} = \sigma^{-2}_{t+1}\mathbf{F}^{-1}_{t+1}\mathbf{X}^T_{\mathcal{A}_{t+1}}\mathbf{y},$$

$$\mathbf{F}_{t+1} = \sigma^{-2}_{t+1}\mathbf{X}^T_{\mathcal{A}_{t+1}}\mathbf{X}_{\mathcal{A}_{t+1}} + \sigma^{-2}_{t+1}g^{-1}_{t+1}\mathbf{X}^T_{\mathcal{A}_{t+1}}\mathbf{X}_{\mathcal{A}_{t+1}} + \lambda \mathbf{I}_{k_{t+1}}$$

Iterative simulation

Finally, the overall proposal distribution can be written as

$$q(\beta_{t+1}, \mathcal{A}_{t+1}, g_{t+1}, \sigma^2_{t+1} | \mathcal{A}_t, g_t, \sigma^2_t, \mathbf{X})$$

$$= q(\beta_{t+1} | \mathcal{A}_{t+1}, g_{t+1}, \sigma^2_{t+1}, \mathbf{X}_{\mathcal{A}_{t+1}}) q(\mathcal{A}_{t+1} | \mathcal{A}_t) q(\sigma^2_{t+1} | \sigma^2_t) q(g_{t+1} | g_t)$$

One can easily simulate from the overall proposal by iteratively simulating from the factors, from right to left and conditioning on the values sampled up to the current step of execution.

Our MH algorithm **converges comparatively fast**.

Accuracy measures for comparison with other methods:

$$\text{MSE} = \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} (y_i - \widehat{y}_i)^2, \quad \text{MAD} = \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} |y_i - \widehat{y}_i|$$

where n_{te} = cardinality of test dataset and $\widehat{y}_i$ = predicted target values

Comparison with competitors

For obtaining a prediction $\widehat{y}^*$ corresponding to a test sample $\mathbf{x}^*$, an estimate of the expected value of the posterior predictive distribution $\mathbb{E}(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X})$ is used. Using MC -integration this expected value can be estimated as follows:

$$\mathbb{E}(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}) = \approx \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{*T} \beta_i$$

For the Bayesian comparison models the way the predictions are computed depend on the output provided by the R-packages:

- R-function *blasso* in package *monomvn* for B. Lasso
- R-function *brq* in package *Brq* for B. ad. Lasso
- R-function *EBglmnet* in package *EBglmnet* for B. el. net
- *horseshoe* and *bayesreg* for horseshoe and horseshoe+, resp.
- *EMVS* and *varbvs* for spike and slab via EM and VI, resp.

Data sets

Real-world studies: The diabetes data set (Efron 2004), see R-package *care*
Predictors: age, sex, body mass index, average blood pressure, and six blood serum measurements, measured from $n = 442$ diabetes patients.
Target variable: quantitative measure of disease progression one year after baseline.
Burn-in: $10,000$ samples are deleted, Thinning: every 10-th one is deleted.
For each of the observed Bayesian models $50,000$ (dependent) samples are generated, except for the Bayesian adaptive lasso where $70,000$ ones are produced. This results in $4,000$ i.i.d. samples each, except of $6,000$ samples for the adaptive lasso. Performing a 5-fold cross-validation, the proposed approach achieves the lowest MMSE as well as the lowest MMAD and thus performs better than all methods under comparison.

MMSE and MMAD

Method	MMSE	MMAD
Our approach	0.4873534	0.5678801
Lasso	0.492067	0.571596
Adaptive lasso	0.4939229	0.5736721
Elastic net	0.4922686	0.5706994
Bayesian lasso	0.4924316	0.5736084
Bayesian adaptive lasso	0.4997307	0.5786672
Bayesian elastic net	0.4895844	0.5727555
Horseshoe	0.4903684	0.5711527
Horseshoe+	0.4919946	0.5727804
Spike and slab (VI)	0.5179594	0.5894747
Spike and slab (EM)	0.4893634	0.568348

Simulated Data studies

Simulated data corresponding to two different artificial models, on purpose including many correlated predictors, from which only a small subset is predictive.
$\Rightarrow$ Difficult variable selection problems

From both artificial models 100 datasets are simulated
Sampling with $n = 50$, $n = 100$ and $n = 200$ training observations.
The number of test observations is always the same: $n_{te} = 200$.

$$y = \beta_1 x_1 + ... + \beta_{100} x_{100} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 1)$$
$$(x_1, ..., x_{100})^T \sim \mathcal{N}(\mathbf{0}, \Sigma)$$
$$\text{diag}(\Sigma) = \mathbf{1}, \quad \Sigma_{i,j} = 0.6 \text{ for } i \neq j$$

with $(\beta_2, \beta_{11}, \beta_{21}, \beta_{51}, \beta_{71}, \beta_{81}) = (-2.5, -2, -1.5, 1.5, 2, 2.5)/\sqrt{3}$ and the remaining coefficients equal to zero.

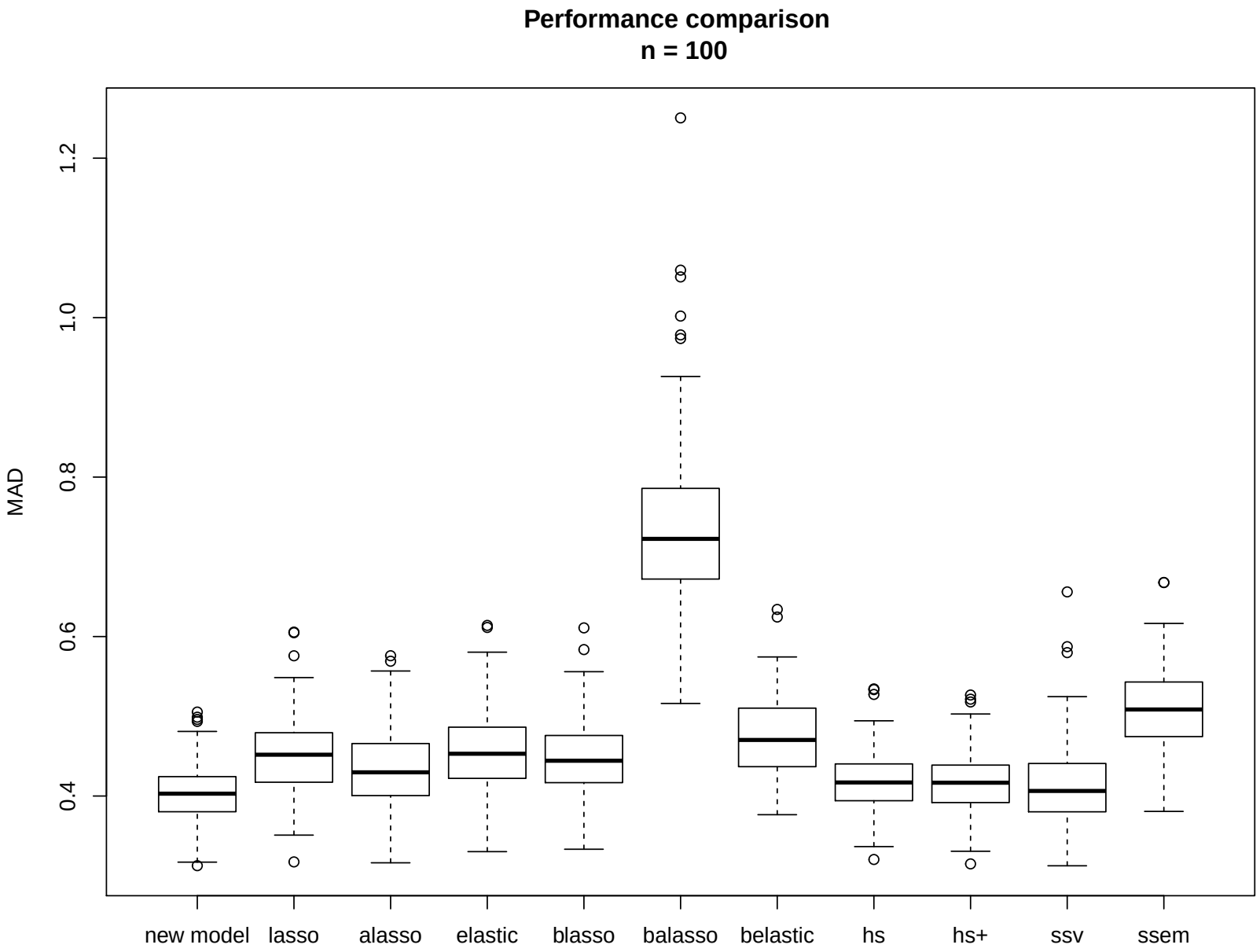
Model is inspired by those used to evaluate the performance of the spike-and-slab lasso in Ročková and George (2018).

Performance comparison

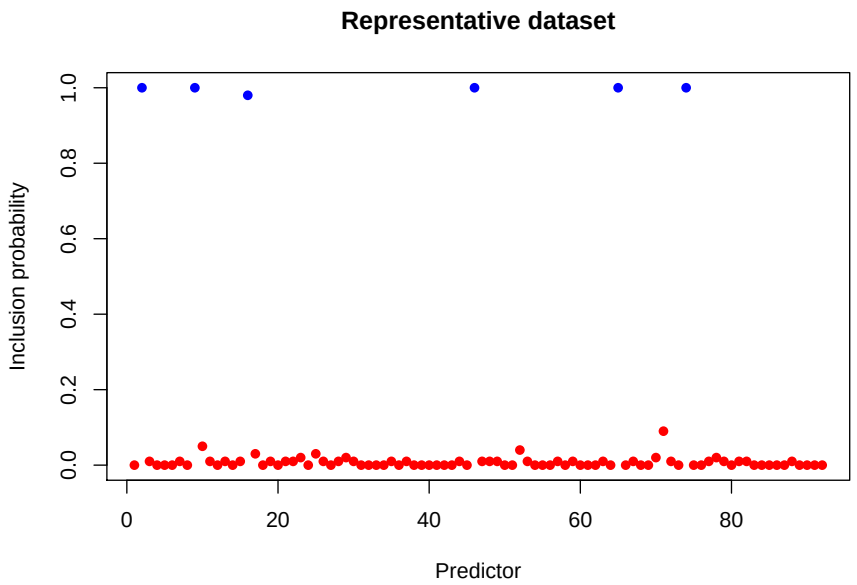
Performance comparison for different specifications of n, MMSE based on a 100 simulated datasets

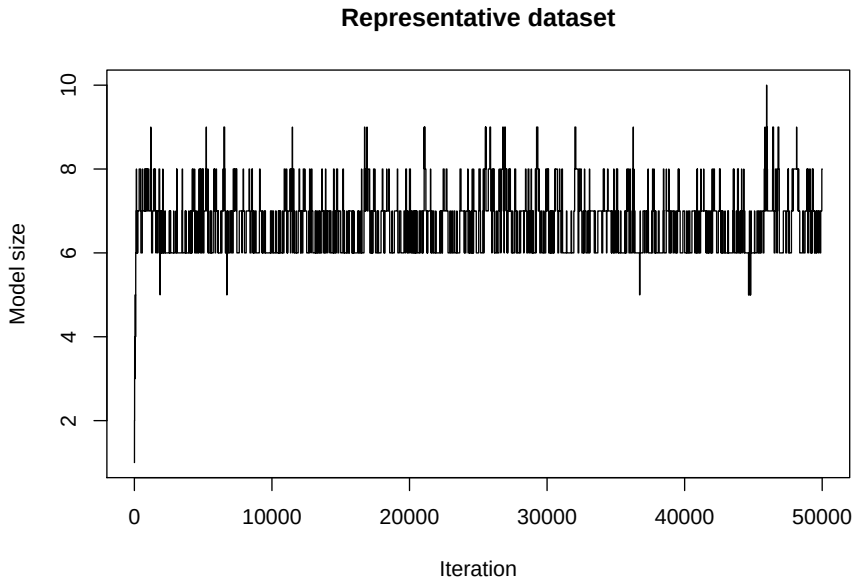
Method	MMSE $n = 50$	MMSE $n = 100$	MMSE $n = 200$
Our approach	0.3619324	0.2496664	0.2405002
Lasso	0.4643155	0.3114765	0.2696669
Adaptive lasso	0.4261128	0.2869056	0.2653867
Elastic net	0.4681363	0.3193174	0.2727069
Bayesian lasso	0.6445547	0.3078518	0.2665262
Bayesian adaptive lasso	0.7616539	0.8052745	0.3911191
Bayesian elastic net	0.777284	0.3367692	0.273061
Horseshoe	0.4305569	0.2710609	0.2512771
Horseshoe+	0.427505	0.2699478	0.2517378
Spike and slab (VI)	0.6992991	0.2563254	0.2462369
Spike and slab (EM)	0.4947604	0.4078821	0.3596638

MAD - highly correlated predictors



Posterior Model Inclusion





Computing time

Model training computation time in seconds, for one train/test split

Method	Diabetes	Sim1 $n = 200$	Sim2 $n = 200$
Our approach	10.7	7.44	27.3
Lasso	0.10	0.13	0.57
Adaptive lasso	0.15	0.27	0.84
Elastic net	1.21	2.08	9.58
Bayesian lasso	1.84	16.0	116.
Bayesian adaptive lasso	87.2	236.	
Bayesian elastic net	1.88	14.6	594.
Horseshoe	9.96	93.8	346.
Horseshoe+	9.13	41.9	597.
Spike and slab (VI)	0.14	0.45	1.67
Spike and slab (EM)	0.01	0.01	0.05

IV. Bayesian Deep Learning

Popularity of **Deep Learning** is increasing rapidly: excellent results in many fields of applied machine learning, including computer vision and natural language processing

Excellent overview in Goodfellow, Bengio and Courville:
Deep Learning. MIT Press 2016

Note: Deep NNs act as Gaussian Processes, see Lee et al. 2018

Bayesian DL overcomes drawbacks of classical DL:

- Network parameters are treated as random variables
- Uncertainty regarding parameters is directly translated into uncertainty about predictions
- Robustness to overfitting (built-in **regularization**)

We need, however, **ABC** methods to compute **posteriors**

- Laplace approximation
- Variational inference, usually with independent Gaussians

Note: Dropout regularization (Gal and Ghahramani 2015) acts like VI

Measuring Uncertainty in Deep Neural Networks

Novel approach for training DNNs using Bayesian techniques presented in
K. Posch and J. Pilz: Correlated Parameters to Accurately Measure Uncertainty in Deep Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, Vol. 32 (2021) No. 3, 1037 - 1051

Our novelty comprises

- variational distribution as product of multiple multivariate normals with **tridiagonal** covariance matrices
- correlations are assumed to be identical $\Rightarrow$ only a few additional parameters need to be optimized

Rationale: Dependent tridiagonal (instead of diagonal only) Gaussians effect an exchange of information between NN layers and neurons

Also, our approach allows an easy evaluation of model uncertainty and is robust to overfitting

Prediction uncertainty

Note: Variational Bayes is just a specific case of local α-divergence minimization:
α-divergence between two densities $p(\mathbf{w})$ and $q(\mathbf{w})$ is given by

$$D_\alpha(p(\mathbf{w})||q(\mathbf{w})) = \frac{1}{\alpha(1-\alpha)}\left(1 - \int p(\mathbf{w})^\alpha q(\mathbf{w})^{(1-\alpha)}\, d\mathbf{w}\right)$$

α-divergence converges for $\alpha \to 0$ to the Kullback-Leibler (KL) divergence typically used in variational Bayes. We used a product of tridiagonal Gaussians as variational density $q(\cdot)$. Moreover, note that

Prediction uncertainty = Epistemic (model) uncertainty
+ Aleatoric (observational) uncertainty

Let **W** denote the rv covering all parameters (weights and biases) of a given neural net **f**. Further, let $p(\mathbf{w})$ denote the prior regarding **W**. According to the Bayes's theorem the **posterior** distribution of **W** is given by the density

KL divergence

$$p(\mathbf{w}|\mathbf{y},\mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{w},\mathbf{X})p(\mathbf{w})}{\int p(\mathbf{y}|\mathbf{w},\mathbf{X})p(\mathbf{w})\, d\mathbf{w}}$$

where $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_\beta\}$ denotes a set of training examples and $\mathbf{y} = (y_1, ..., y_\beta)^T$ holds the corresponding class labels.
Note that $p(\mathbf{y}|\mathbf{w},\mathbf{X}) = \prod_{i=1}^{\beta} \mathbf{f}(\mathbf{x}_i;\mathbf{w})_{y_i}$

The integral above is commonly intractable due to its high dimension β. Variational inference aims at approximating the posterior with the so-called **variational density** $q_\phi(\mathbf{w})$. The variational parameters ϕ are optimized by minimizing the KL divergence

$$D_{KL}(q_\phi(\mathbf{w})||p(\mathbf{w}|\mathbf{y},\mathbf{X})) = \mathbb{E}_{q_\phi(\mathbf{w})}\left(\ln \frac{q_\phi(\mathbf{w})}{p(\mathbf{w}|\mathbf{y},\mathbf{X})}\right)$$

Since the posterior is unknown this divergence cannot be minimized directly. However, minimization of D_{KL} is equivalent to the minimization of so-called

ELBO

negative log **evidence lower bound**

$$L_{VI} = -\mathbb{E}_{q_\phi(\mathbf{w})}\left[\ln p(\mathbf{y}|\mathbf{w},\mathbf{X})\right] + D_{KL}(q_\phi(\mathbf{w})||\textit{prior } p(\mathbf{w}))$$

Commonly, mini-batch gradient descent is used for optimization
To take account of the resulting reduction of the number of training examples used in each iteration of the optimization, the rescaling is necessary: in the k-th iteration we minimize

$$\widehat{L}_{VI} = -\frac{1}{m}\sum_{i=1}^{m}\left\{\ln \mathbf{f}(\tilde{\mathbf{x}}_i;\mathbf{w}_k)_{\tilde{y}_i}\right]\} + \frac{1}{\beta}D_{KL}(q_\phi(\mathbf{w})||p(\mathbf{w}))$$

where $\mathbf{w}_k$ denotes a sample from $q_\phi(\mathbf{w})$, m is mini-batch size, and $\tilde{\mathbf{x}}_1, ..., \tilde{\mathbf{x}}_m, \tilde{y}_1, ..., \tilde{y}_m$ denote the mini-batch itself. Summing up:

- Frequentist deep learning penalizes (Euclidean) norm of $\mathbf{w}$
- Bayesian deep learning penalizes deviations of the variational distribution from the prior.
  Crucial difference: **sampled** network parameters.

Posterior predictive distribution

In Bayesian deep learning predictions are based on the posterior predictive distribution, i.e. the distribution of a class label y^* for a given example $\mathbf{x}^*$ conditioned on the observed data $\mathbf{y}, \mathbf{X}$:

$$p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}) = \int p(y^*|\mathbf{w}, \mathbf{x}^*) p(\mathbf{w}|\mathbf{y}, \mathbf{X}) \, d\mathbf{w}$$

This distribution can be approximated via MC integration

$$p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}) \approx \frac{1}{N} \sum_{i=1}^{N} \mathbf{f}(\mathbf{x}^*; \mathbf{w}_i)_{y^*}$$

where $\mathbf{w}_1, ..., \mathbf{w}_N$ denote samples from $q_\phi(\mathbf{w})$

Implementation of prior: In each layer $j = 1, ..., d$ we sample from rvs

$$\mathbf{W}_j = \mathbf{m}_j + \mathbf{L}_j \mathbf{X}_j \quad \text{with} \quad \mathbf{X}_j \sim \mathcal{N}(\mathbf{0}_{K_j}, \mathbf{I}_{K_j})$$

and define the variational d. of $\mathbf{W}_j$ as multivariate normals

$$\mathbf{W}_j \sim \mathcal{N}(\mathbf{m}_j, \Sigma_j), \quad \Sigma_j = \mathbf{L}_j \mathbf{L}_j^T$$

Tridiagonal Gaussians

Choosing

$$\mathbf{L}_j := \begin{pmatrix} a_{j1} & & & & \\ c_{j1} & a_{j2} & & & \\ & c_{j2} & a_{j3} & & \\ & & \ddots & \ddots & \\ & & & c_{j,K_j-1} & a_{jK_j} \end{pmatrix}$$

we end up with a **tridiagonal** cov. matrix Σ_j with **equal correlations**

To train a neural net $\mathbf{f}(\cdot, \mathbf{w})$ we need the partial derivatives of the approximation $\widehat{L}_{VI}$ with respect to all variational parameters.

In particular, the partial derivatives of the loss function L used in deep learning and the partial derivatives of $D_{KL}(q||p)$ have to be computed.

Note: Loss function L = negative log likelihood of the data, i.e.

- L = cross-entropy loss in case of classification and
- L = Euclidean loss in case of regression.

Implementation

We have implemented the proposed approach by modifying and extending the popular open-source
Deep Learning framework **Caffe** (Jia et al. 2014). Up to now we have not parallelized our code such that it can run on GPU.

The **Pseudocode** of our implementation is presented in our paper. The code shows how a classical, i.e. frequentist, inner product, or convolutional layer can be extended in order to fit with our methodology.

Performance evaluation
Comparison includes the frequentist approach, the proposed approach without correlations (see Steinbrener, Posch and Pilz 2020), and, finally, the popular approach which applies dropout before every weight layer in terms of a Bernoulli variational distribution (Gal and Ghahramani 2015).

Criteria: prediction accuracy and quality of the uncertainty information

MNIST benchmark

Benchmark datasets: MNIST (Deng 2012) and CIFAR-10 (Krizhevsky 2009)
training set: 60000 grayscale images of digits, test set: 10000 images
200 samples from corresp. variational distrib. per test image

TABLE
TEST ERRORS OF THE TRAINED MODELS

Model	Neurons	Test error
Frequentist	100	0.84%
Gauss cor.	100	0.70%
Gauss ind.	100	1.05%
Bernoulli	100	0.78%
Frequentist	250	0.70%
Gauss cor.	250	0.61%
Gauss ind.	250	1.00%
Bernoulli	250	0.78%

Uncertainty evaluation

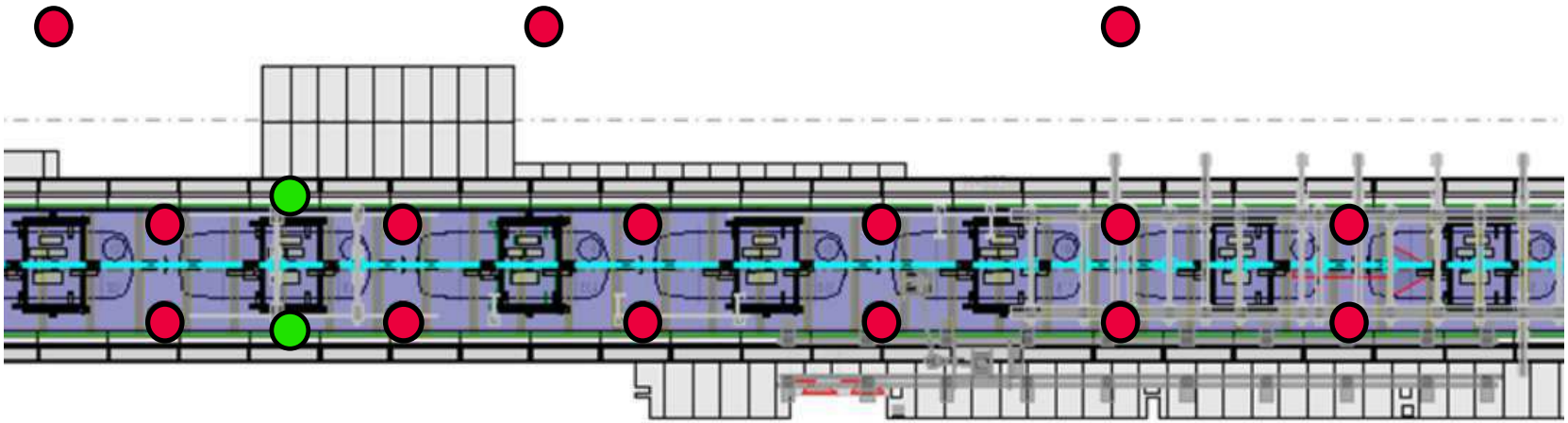
Evaluation of overall quality of uncertainty information based on two measures: log-L and Brier score

TABLE
LOG-LIKELIHOOD AND BRIER SCORE OF THE TESTING DATASET

Model	Neurons	Log-likelihood	Brier score
Gauss cor.	100	-251.7366	0.01128257
Gauss cor.	250	-220.9156	0.01041743
Gauss ind.	100	-377.5755	0.0160266
Gauss ind.	250	-336.4502	0.01522011
Bernoulli	100	-302.7554	0.01338242
Bernoulli	250	-270.3255	0.01253694

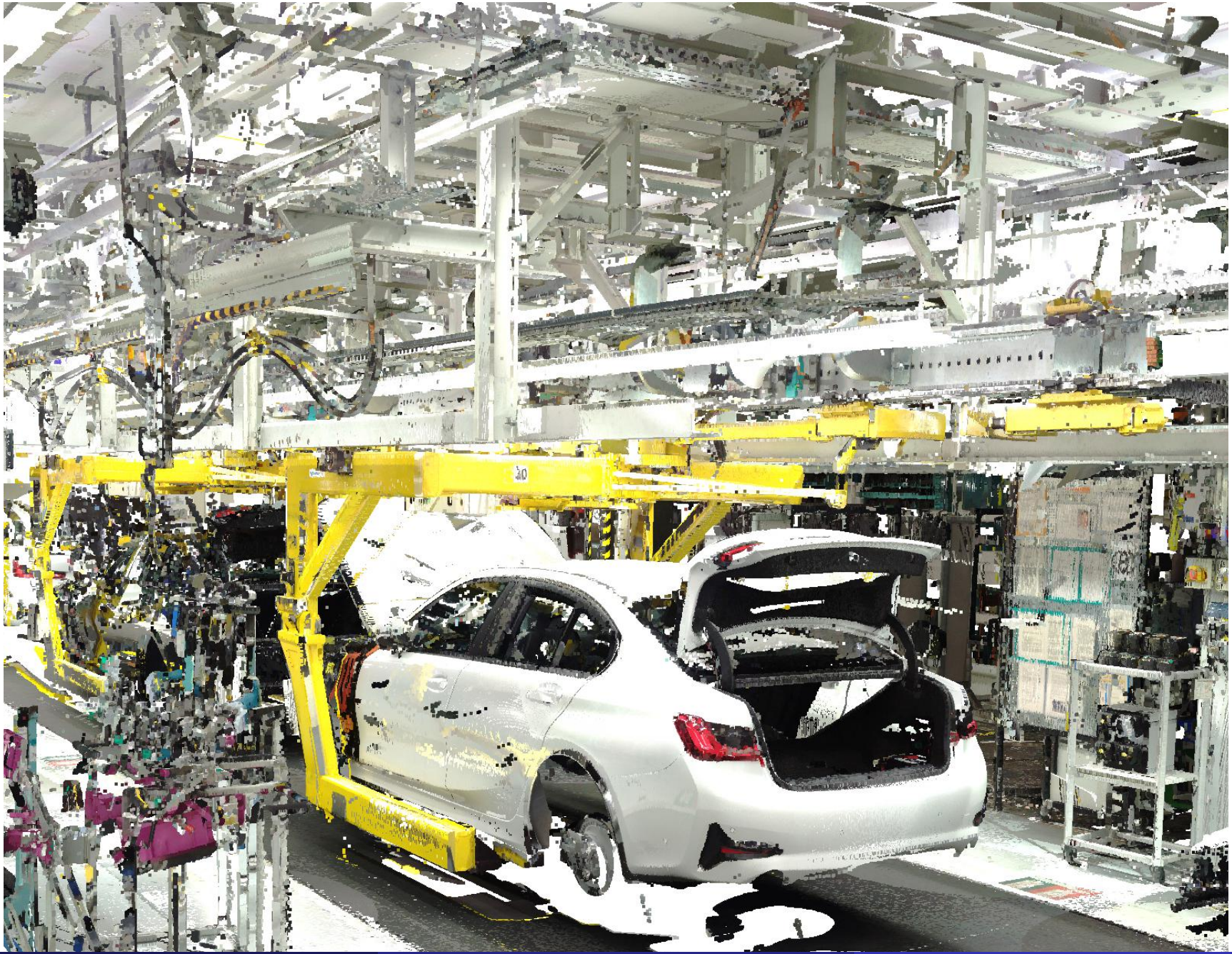
3D Point Clouds

Modification: Bayes Deep Learning for 3D point cloud segmentation
Application in Automotive Industry, BMW Group Munich-Germany
Two recent publications in MDPI journals ”Entropy 2021" and "Modelling 2021"
Joint work with my youngest PhD Christina Petschnigg

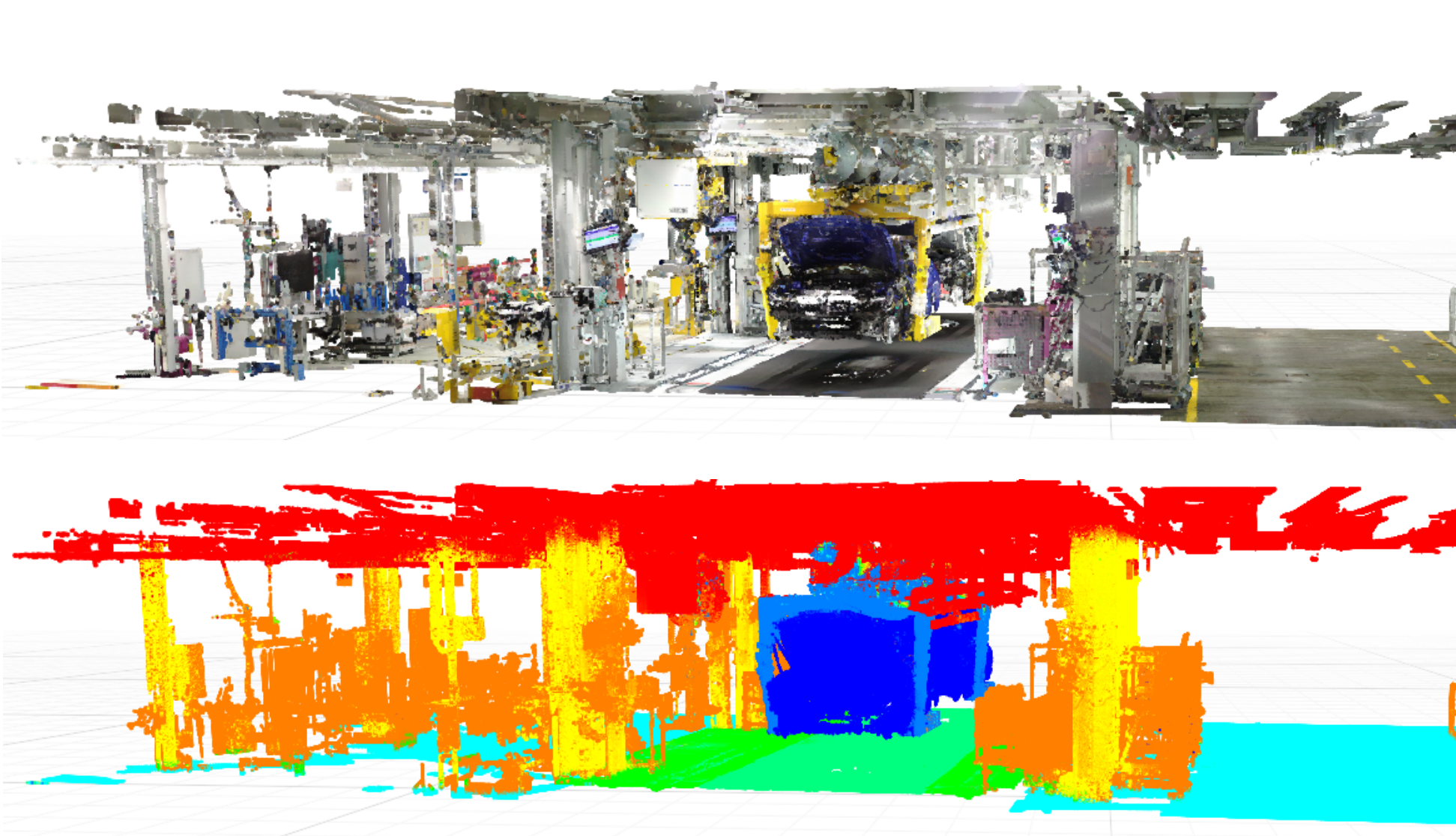


Location of the laser scanner
Additional location of the laser scanner

BMW Assembly Line



Data recording



Classes: Car, Hanger, Floor, Band, Lineside, Wall, Column, Ceiling, Clutter

Thank you and all the best for the further growth of Statistics and Data Science in Pakistan!

Please, also have a look at our most recent contributions to merging ideas from Machine Learning and Bayesian Statistics:

Anna Jenul, Stefan Schrunner, Jürgen Pilz and Oliver Tomic:
A user-guided Bayesian framework for ensemble feature selection in life science applications (UBayFS). Machine Learning, August 2022 https://doi.org/10.1007/s10994-022-06221-9

Konstantin Posch, Christian Truden, Philipp Hungerländer and Jürgen Pilz: A Bayesian approach for predicting food and beverage sales in staff canteens and restaurants. Int. Journal of Forecasting 38 (2022), 321-338

References

O. Roustant, D. Ginsbourger, Y. Deville, DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization, Journal of Statistical Software 51 (2012) 1, 1–55

H. Kazianka, J. Pilz, Objective Bayesian analysis of spatial data with uncertain nugget and range parameters, The Canadian Journal of Statistics 40 (2012), 304–327

M. Gu, X. Wang, J. O. Berger, Robust Gaussian stochastic process emulation, Annals of Statistics 46 (2018) 6A, 3038 - 3066

M. Gu, J. Palomo, J.O. Berger, RobustGaSP: Robust Gaussian Stochastic Process Emulation in R. The R Journal Vol. 11/01, June 2019

N. Vollert, M. Ortner and J. Pilz, Robust Additive Gaussian Process Models Using Reference Priors and Cut-Off-Designs. J. Applied Mathematical Modelling 65 (2019), 586-596

References

J. Steinbrener, K. Posch and J. Pilz. Variational Inference to Measure Model Uncertainty in Deep Neural Networks. Sensors 2020, 20, 6011; doi:10.3390/s20216011

K. Posch, M. Arbeiter and J. Pilz, A novel Bayesian approach for variable selection in linear regression models. Computational Statistics and Data Analysis 144 (2020), 106881

Alhamzawi, Rahim and Taha Mohammad Ali, Haithem. The Bayesian adaptive lasso regression. Math. Biosciences 303(2018), 75 - 82

R. Tibshirani, Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological) 58 (1996) 1, 267–288

T. Park and George Casella, The Bayesian Lasso. Journal of the American Statistical Association 103 (2008) 482, 681 - 686

References

Hui Zou, The Adaptive Lasso and Its Oracle Properties. Journal of the American Statistical Association 101 (2006) 476, 1418-1429

Ch. Leng, Tran, Minh-Ngoc and D. Nott, Bayesian adaptive Lasso. Annals of the Institute of Statistical Mathematics 66 (2014) 2, 221 - 244

Zou, Hui and T. Hastie, Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67 (2005) 2, 301 - 320

A. Huang, Shizhong Xu and Xiao-hui Cai, Empirical Bayesian elastic net for multiple quantitative trait locus mapping. Heredity 114 (2015), 107 - 115

M. Wang, Sun, Xiaoqian and Lu, Tao, Bayesian structured variable selection in linear regression models. Computational Statistics 30 (2015) 1, 205 - 229

References

A. Zellner, On Assessing Prior Distributions and Bayesian Regression Analysis With g-Prior Distributions. P. K. Goel and A. Zellner, Eds., Basic Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti, 1986, 233 - 243

Meïli C. Baragatti and D. Pommeret, A study of variable selection using g-prior distribution with ridge parameter. Computational Statistics & Data Analysis 56 (2012), 1920-1934

R.B. Gramacy, monomvn: Estimation for Multivariate Normal and Student-t Data with Monotone Missingness, 2018, R package version 1.9-8, https://CRAN.R-project.org/package=monomvn

Rahim Alhamzawi, Brq: An R package for Bayesian Quantile Regression, Working Paper, 2018, https://cran.r-project.org/web/packages/Brq/Brq.pdf

J. Friedman, T. Hastie and R. Tibshirani, Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software 33 (2010) 1, 1–22

References

B. Efron, T. Hastie, Trevor, I. Johnstone and R. Tibshirani, Least angle regression. Ann. Statist. 32 (2004) 2, 407–499

V. Zuber and K. Strimmer, care: High-Dimensional Regression and CAR Score Variable Selection. R package version 1.1.10, 2017, https://CRAN.R-project.org/package=care

J. Lokhorst, B. Venables and B. Turlach, lasso2: L1 constrained estimation aka 'lasso'. R package version 1.2-19, 2014, https://CRAN.R-project.org/package=lasso2

V. Ročková and E. I. George, The Spike-and-Slab LASSO. Journal of the American Statistical Association 113 (2018) 521, 431 - 444

Anhui Huang and Dianting Liu, EBglmnet: Empirical Bayesian Lasso and Elastic Net Methods for Generalized Linear Models. R package version 4.1, 2016, https://CRAN.R-project.org/package=EBglmnet

References

C. M. Carvalho, N. G. Polson and J. G. Scott, The horseshoe estimator for sparse signals. Biometrika 97 (2010) 2, 465-480

E. Makalic and D. Schmidt, High-Dimensional Bayesian Regularised Regression with the BayesReg Package. arXiv:1611.06649v3, 2016

N. G. Polson and J. G. Scott, Local shrinkage rules, Levy processes and regularized regression. Journal of the Royal Statistical Society (Series B) 74 (2012) 2, 287-311

A. Bhadra, J. Datta, N.G. Polson and B. Willard, Brandon, The Horseshoe+ Estimator of Ultra-Sparse Signals. Bayesian Analysis 12 (2017) 4, 1105–1131

P. Carbonetto and M. Stephens, Scalable Variational Inference for Bayesian Variable Selection in Regression, and Its Accuracy in Genetic Association Studies. Bayesian Analysis 7 (2012) 1, 73–108

References

V. Ročková and E. I. George, EMVS: The EM Approach to Bayesian Variable Selection. Journal of the American Statistical Association 109 (2014) 506, 828-846

A. Bhattacharya, D.Pati, N. S. Pillai and D. B. Dunson, Dirichlet–Laplace Priors for Optimal Shrinkage. Journal of the American Statistical Association 110 (2015) 512, 1479-1490

Zhang, Yan and H. D. Bondell, Variable Selection via Penalized Credible Regions with Dirichlet–Laplace Global-Local Shrinkage Priors. Bayesian Analysis 13 (2018), 823–844

Chen, Su and S. G. Walker, Fast Bayesian variable selection for high dimensional linear models: Marginal solo spike and slab priors. Electronic Journal of Statistics 13 (2019) 1, 284–309

N.G. Polson and S.L. Scott, Data augmentation for support vector machines. Bayesian Analysis 6 (2011) 1, 1–23

References

Zhou, Quan and Y. Guan, Fast Model-Fitting of Bayesian Variable Selection Regression Using the Iterative Complex Factorization Algorithm. Bayesian Analysis 14 (2019) 2, 573–594

Ch. Petschnigg and J. Pilz: Uncertainty Estimation in Deep Neural Networks for Point Cloud Segmentation in Factory Planning. Modelling 2021, 1, 1-17

J. R. Hershey and P. A. Olsen, Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models. IEEE International Conference on Acoustics, Speech and Signal Processing 4 (2007), 317–320

Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv:1408.5093, 2014

A. Krizhevsky, I. Sutskever, Ilya and G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems 25 (2012), 1097-1105

References

Y. Gal, Yarin and Z. Ghahramani, Zoubin, Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. Proceedings of The 33rd International Conference on Machine Learning 2015, 1050–1059

L. Deng, The MNIST Database of Handwritten Digit Images for Machine Learning Research. IEEE Signal Processing Magazine 29 (2012), 141-142

A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images. 2009

H. M. D. Kabir, A. Khosravi, M. A. Hosen and S. Nahavandi, Neural Network-Based Uncertainty Quantification: A Survey of Methodologies and Applications. IEEE Access 6 (2018), 36218-36234

Ch. Petschnigg, M. Spitzner, L. Weitzendorf and J. Pilz, From a Point Cloud to a Simulation Model—Bayesian Segmentation and Entropy Based Uncertainty Estimation for 3D Modelling. Entropy 23 (2021) 301, 1 - 27