Ludwig Maximilian University of Munich
Faculty of Mathematics,
Informatics and Statistics

Department of Statistics
Bachelor Thesis

An Analysis of Bavarian Settlement Areas based on Neighbourhood Information using Gaussian Processes

Supervising Professor : Prof. Dr. Volker Schmid

Supervisor : Christopher Küster

Author : Benedikt Arnthof

Datum : April 29, 2020

Abstract

This thesis builds on the methods presented in [43] that were previously applied in a seminar on geocomputation with R. The main goal is to describe and analyze the effects of habitat covariates, such as average temperature or elevation, as well as geographic location information on the chances of finding traces of human iron age settlements in modern day Bavaria. First, the data acquisition and preprocessing is described briefly. Section 3 introduces the logistic model as an expansion of the linear model, as it is ideal to describe the binary data at hand. Next, various techniques for smoothing and interpolation are discussed. Here the main focus lies on Generalized Additive Models [53] and Gaussian Processes and their use in Kriging [28] or spatial interpolation. Further, it is described how the choice of the covariance function can influence smoothing results and, how the use of latent Gaussian Processes in combination with their spectral density representation can dramatically increase the sampling speed in a bayesian setting. The bayesian methods are then described in section 5. This section also discusses the advantages and limitations of various R [41] interfaces to Stan [46], the sampler chosen for this endeavour. The model specifications, and results are presented in section 6, coupled with predictive maps, and a comparison of the different covariance kernels used. Section 7 further discusses these results, and reviews additional methods for decreasing computational load in Gaussian Process Regression.

Contents

1 Motivation

The geographic data that are investigated in this work have previously been used to compare the performance of simple logistic regression, support vector machines and a random forest classifier, in predicting the probability of finding remains of human settlements of the iron age in Bavaria. Because the treatment of spatial data like ordinary non-spatial data can lead to overoptimistic results in predictive accuracy of models [8], these effects were accounted for using spatial repeated cross validation. This way of partitioning the data into spatially coherent pieces proved to drastically increase the variance of the performance estimates, and did not incorporate the spatial information present into the models. As a result, this thesis combines logistic regression with "Kriging", a technique to model spatial covariance first introduced by Danie Krige in 1951. [28] Because this way of spatial interpolation, also referred to as "Gaussian Process Regression" outside of geostatistics, is closely connected to Bayesian inference, all of the approaches of modeling spatial structure present in data are also applied in Stan.
The influence of the chosen covariance kernel was also of interest. A comparison based on the predictive power of the models incorporating these functions is given, alongside the predictive maps generated by them. The advantages and drawbacks of all methods were compared, as both flexibility in modeling, computational complexity and general ease of use were of interest.

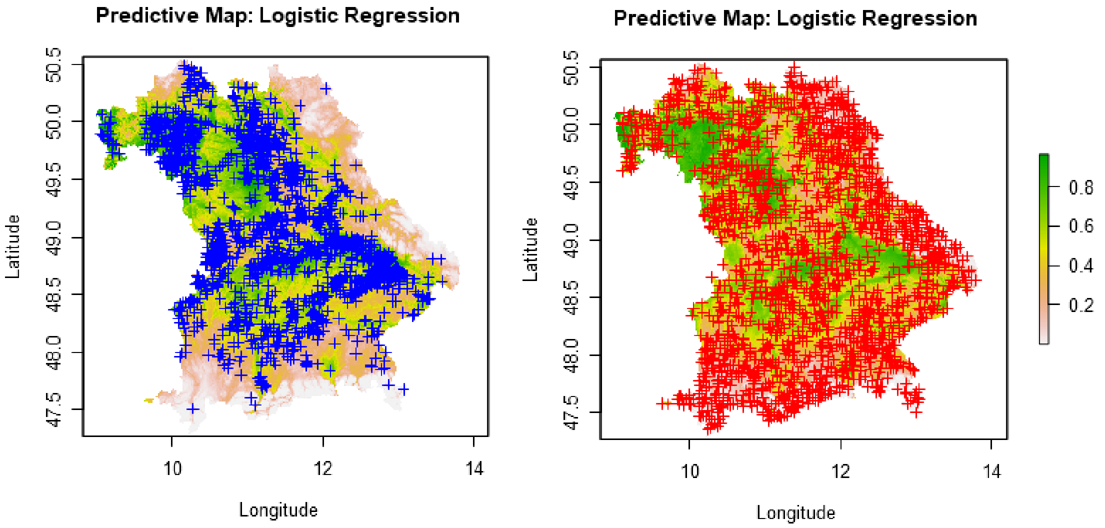


Figure 1: Two samples of 2000 points each. Left: The positions of 2000 locations where traces of human Iron Age sites were found. Right: 2000 non-site points that were sampled randomly as described in section 2. The background colors indicate the prediction of a simple logistic regression with only linear terms. Mean AUC value for this model was 0.76.

2 Data – Access & Preprocessing

This section illustrates the data acquisition and preprocessing steps that were taken to generate the data used for both modeling, and predictive mapping respectively.

2.1 Prehistoric Bavaria

The data used in this analysis were extracted from a database that was assembled by Peer Fender. The aim of his work was a GIS-assisted archaeological landscape evaluation of prehistoric settlement development using Bavaria as an example. The attempt to obtain detailed information regarding different aspects of Neolithic, Iron-, and Bronze Age settlement structure was made using only openly available data. [19]
This database is available to students of the LMU [17] and serves various educational projects. Both coordinates of archaeological sites, class information like type and epoch, and meteorological data, such as annual average precipitation, are provided. To make the results comparable to the previous work on prediction of sites, and limit the run time of the Bayesian models, the data set was restricted to 6111 sites that were classified as Iron Age sites. This included both discoveries from the "Hallstatt" (800 - 450 BCE) , and "La Tène" (450 BCE - 50 CE) cultures.
Variables of interest included elevation, distance to the closest source of water, average annual frost days, temperature, precipitation, sun hours, as well as the coordinates of the sites themselves. Additionally, the inclination and rotation of slopes and the Topographic Position Index (TPI) were also chosen for modeling purposes, as they seemed appropriate to model the habitat suitability for Iron Age people. The influences of local climate and elevation on the choice of settlement location are still relevant even in modern times. When speaking about the distance to the nearest water source it should be clarified, that this refers to the shortest direct way from the point on the map to the next larger body of water. As techniques for building and maintaining wells were already widely known prior to the iron age, these distances may seem less relevant at first, but larger bodies of water such as the lakes in the south of Bavaria and the Danube river provide other resources, like fish, and were of interest because of this.

2.2 Preprocessing

Because only locations where archaeological findings were confirmed were available in the data, an additional random sample of 6111 "Non-sites" was drawn from Bavaria. To keep everything consistent with previous work on the data set, these non-sites were drawn from all possible locations under the restriction that they must not fall closer than 1500 meters to the sites. 1500 meters seemed to be a suitable compromise between potentially invalidating the locations of known sites by letting non-sites fall next to them, and introducing artificial bias into the data by sampling non-sites too far away from sites. The vast majority of sites was found near the center region of Bavaria. Restricting non-sites to locations too far away from these regions would push them off to either the lower regions of north west Bavaria or the southern and eastern regions that are dominated by the Alps and and mountain ranges of the Bavarian Forest respectively. That would artificially inflate the influence of the elevation variable on the prediction of the model.

Sampling non-sites randomly also implies that the underlying basis for sites and non-sites must be the same. Because the original measurements included in the data set by Fender could not be reproduced for the sampled non-sites, the final data set used for the specification of the models was constructed from scratch. All covariate values at the locations of interest were extracted from publicly available raster data. The "raster" package [42] allows importing digital elevation models at various resolutions. A resolution of 90 meters was chosen, as the values supplied by Fender were average values in a radius of 50 meters around the points of interest. The elevation data is based on a Shuttle Radar Topography Mission in the year 2000, and can be processed to rasters containing inclination, rotation, and topographic position index (TPI) values. Inclination and rotation of slopes are calculated according to the methods described in [23]. First, a gradient estimation is performed on the digital terrain model. Even with limited data, a reasonably detailed hill shading output may be produced through combinations of biased slope estimates, as these combinations yield up to four times as many intermediate values as there are elevation values in the terrain model. [24] A reflectance map is then calculated from the gradient estimate which is in turn corrected and processed for graphic output.

The TPI of each raster cell is equal to the difference of the value of the cell and the average of the 8 neighbouring cells. All of these terrain model calculations can be thought of as applying different convolutions to the elevation "image". Both slope and rotation are given in degrees, but to aid in making the values for rotation interpretable, these values were split into a factor of eight different levels, corresponding to north, north-east, east, south-east, etc.

Similarly, the rasters containing the average yearly temperature- and precipitation values for the thirty year period beginning in 1971 and ending in the year 2000 were imported using the "getData" function of the "raster" package. [20]

Average yearly frost days and average sun hours per day per year were available as TIFF files on the Geoserver of the German Weather Service. [14] [15].

Because no suitable raster containing the minimum distances to the nearest larger body of water was available online, this raster layer has been manually calculated from a shapefile [21] of german rivers and lakes using standard raster processing functionality. First, an empty raster of the correct resolution and projection was created and filled with random noise. Then the reprojected shapefile was added to this layer. Then distances to the cells containing the rivers and lakes were calculated. This procedure did not consider elevation in the distance calculation.

Apart from reprojecting all layers to the same coordinate reference system and adding them together to form a predictor stack, no further preprocessing was required. The "evidence" data were then extracted from this raster stack at the locations of sites and non-sites alike.

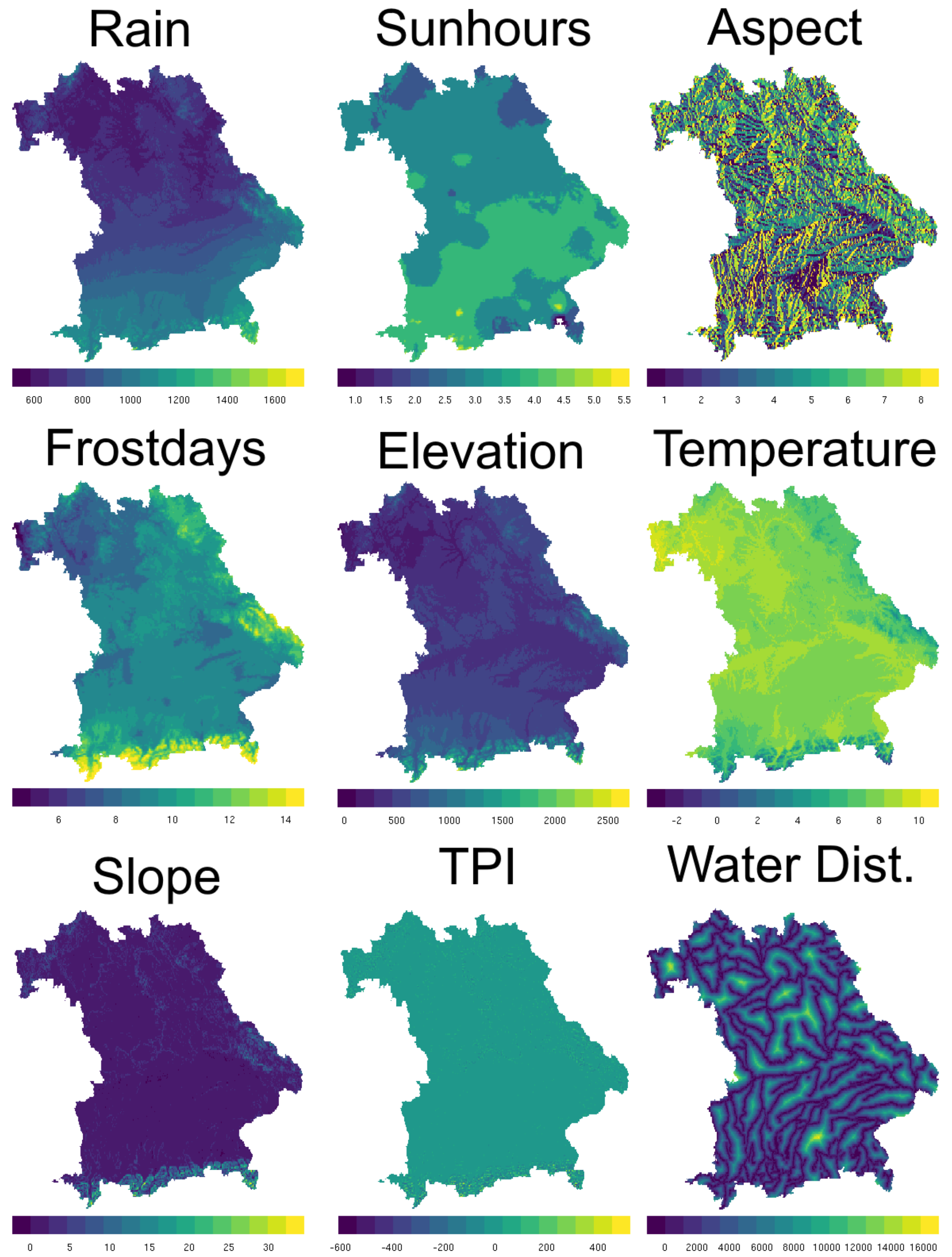


Figure 2: The rasters the dataset was extracted from. Aspect is the 8 class slope rotation. Values for rain, temperature and frostdays are yearly averages. Sunhours is given in average daily hours per year. The distance to the nearest large body of water is given in meters.

3 The Generalized Linear Model

This section introduces logistic regression as a generalization of linear models in the context of exponential families. It is demonstrated that logistic regression can be used as a linear classifier.

3.1 The Exponential Family of Distributions

In the fields of probability theory and statistics, a few key distributions, such as the normal distribution or the binomial distribution are widely used to construct models and make claims about underlying data generating processes. These distributions can be united in a single family of distributions, the Exponential Family of Distributions.

A family of distribuions $\mathbb{P}_{X,\Theta} = \{f(x;\vartheta)|\vartheta \in \Theta\}$ is called a "k-parametric Exponential Family" of distributions if the densities $f(x;\vartheta)$ exist and can be transformed to the representation:

$$f(x;\vartheta) = c(\vartheta)b(x)\exp\left(\sum_{j=1}^{k}\gamma_j(\vartheta)t_j(x)\right)$$

Where $t_j(x)$ is a vector of feature function values and $c(\vartheta)$ can be thought of as a normalization constant.

$\gamma = \gamma(\vartheta)$ is defined as the "natural parameter" of the exponential family, belonging to the natural parameter space $\gamma(\Theta) = \{\gamma(\vartheta)|\vartheta \in \Theta\} \subseteq \mathbb{R}^k$. [22]

θ, the parameter vector, the set of feature functions t, and the carrier density b thus determine the distribution. [30]

The carrier density b is usually chosen based on reasonable assumptions about the data generating process or other domain specific knowledge. The feature functions t are usually specified based on the data. In the simplest case a function t may be the identity function. This is already very flexible, as it allows the inclusion of polynomial terms of features and interactions between variables. After the density b is chosen and the features of interest are specified, the parametervector θ can be calculated with standard maximum likelihood approaches. [30]

As mentioned above, both the Bernoulli and the Binomial distributions are members of the exponential family of distributions. For example, the Bernoulli distribution may be transformed using the indicator function:

$$\begin{aligned} f(x;p) &= p^x(1-p)^{1-x}I_{\{0,1\}}(x) \\ &= \underbrace{(1-p)}_{c(p)}\underbrace{I_{\{0,1\}}}_{b(x)}\exp(\underbrace{x}_{t(x)}\underbrace{\log\left(\frac{p}{1-p}\right)}_{\gamma(p)}) \end{aligned}$$

Further, it can be derived that the log-likelihood of a member of the exponential family has partial derivatives of the form

$$\begin{aligned} -\frac{\partial \log c(\gamma)}{\partial \gamma_j} &= \mathbb{E}_\gamma\left(t_j(X)\right) \\ -\frac{\partial^2 \log c(\gamma)}{\partial \gamma_j \partial \gamma_k} &= \mathrm{Cov}_\gamma\left(t_j(X), t_k(X)\right) \end{aligned}$$

which is useful for calculateing maximum likelihood estimates of θ. Coupled with the fact that the log likelihood of an exponential family is concave, this guarantees that the maximum likelihood estimate for θ is unique.

These properties are used to build upon the linear model to construct the class of "Generalized Linear Models" (GLMs).

3.2 Generalized Linear Models & Logistic Regression

Linear models are very useful for regression analyses where the target variable is continuous and, at least after a suitable transformation, approximately normally distributed. [32]

This can be a limitation when trying to model non-linear effects, as even the inclusion of polynomial terms may be insufficient to describe the intrinsic relationships present in the data. Approaches on how to solve these problems are discussed in the next section.

Additionally, modeling a dichotomous data set, such as archaeological site and non-site data, with a linear model is destined to yield results that are hard to interpret and may prove to be nonsensical. When trying to predict probabilities of a target variable being present at a certain location, for some arbitrary combination of features, the result may be either negative or larger than 1, thus a strictly linear approach is insufficient.

Generalized linear models comprise many different regression approaches for not necessarily normally distributed target variables within a methodologically consistent framework and include, for example, the logit model for binary target variables. [32]

In the case of binary data, the response Y is limited to only two different outcomes, labelled 0 for non-site and 1 for site here. One may formalize this using the notation

$$\mathrm{p}\left(Y_i = 0\right) = 1 - \pi_i; \quad \mathrm{p}\left(Y_i = 1\right) = \pi_i$$

for the probabilities of non-site and site. The goal of a regression analysis is the estimation of the effects of the covariates on these conditional probabilities.

$$\pi_i = \mathrm{P}\left(y_i = 1 | x_{i1}, \ldots, x_{ik}\right) = \mathrm{E}\left(y_i | x_{i1}, \ldots, x_{ik}\right)$$

As mentioned above, limiting the linear predictor

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik} = \boldsymbol{x}_i' \boldsymbol{\beta}$$

to the interval $[0, 1]$ would impose a wide range of restrictions on the parameter estimates $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)'$. Most commonly, these kind of binary regression problems utilize a response function h to transform the linear predictor to a possible range of values of $[0, 1]$. This leads to a representation of the probabilities π_i of the following form:

$$\pi_i = h\left(\eta_i\right) = h\left(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik}\right)$$

Where η_i is the linear predictor term and h is a cumulative distribution function as depicted in figure 3

The logistic model, or logit-model, that was chosen to describe the relationships between site/non-site and the covariates introduced in section 2, uses the logistic response function.

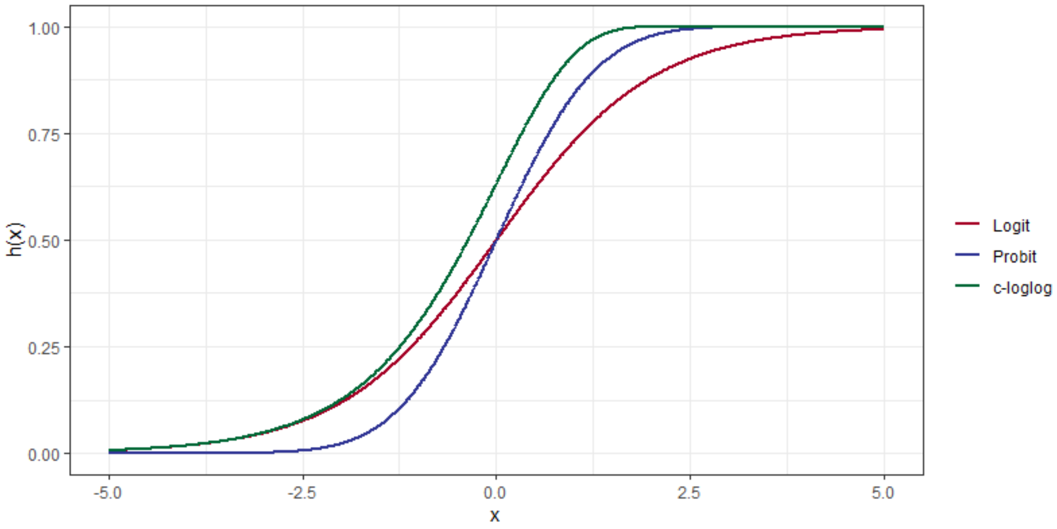


Figure 3: Three different common response functions. The most commonly chosen ones are the logit function for logistic regression and the probit function, the quantile function of the normal distribution. The asymmetric c-loglog function is used when the probabilities of interest are either very small or very large.

$$\pi = h(\eta) = \frac{\exp(\eta)}{1+\exp(\eta)}$$

Solving for the linear predictor η and exponentiating yields the "Odds-representation":

$$\frac{\pi}{1-\pi} = \exp(\beta_0)\exp(\beta_1 x_1) \cdot \ldots \cdot \exp(\beta_k x_k)$$

Thus, the effects of the variables on the odds of the outcome are exponential. This leads to a relatively straightforward interpretation of model coefficients. Positive coefficients can, ceteris paribus, be interpreted as having a positive effect on the odds of the outcome, a rise in the value of a variable with a negative coefficient in turn decreases the odds of the outcome accordingly.

Extending linear models through link functions allows appropriate model specification while still retaining a wide range of flexibility in the linear predictor itself. Polynomial or even non-parametric relationships can be included through customization of the model matrix X. (More on that in the next section.)

Furthermore, it should be mentioned, that when using logistic regression as a method for two-class classification and prediction, the resulting decision boundary is always a linear hyperplane. Thus, while allowing for arbitrary variable transformations within the linear predictor term, logistic regression is still a linear classifier at its core, as depicted in figure 4 The final value for this decision boundary is of course not fixed at 0.5. For a clinical study it may, for example, be the goal to minimize the false negative rate, i.e. to avoid classifying people as healthy when they are actually not. This could be accomplished with a decision limit smaller than 0.5 (if 0 is used to encode "healthy" in the data.) Finding the ideal value for the decision threshold is a challenge that directly motivates a measure of the overall performance of the models. Plotting the true positive rate (TPR) against the false positive rate (FPR) at any decision threshold between 0 and 1 yields the Reciever Operating Curve. (ROC) The total area under this curve can be used as a measure of predictive performance, where $AUC = 0.5$ would be equal to random choice, and $AUC = 1$ would

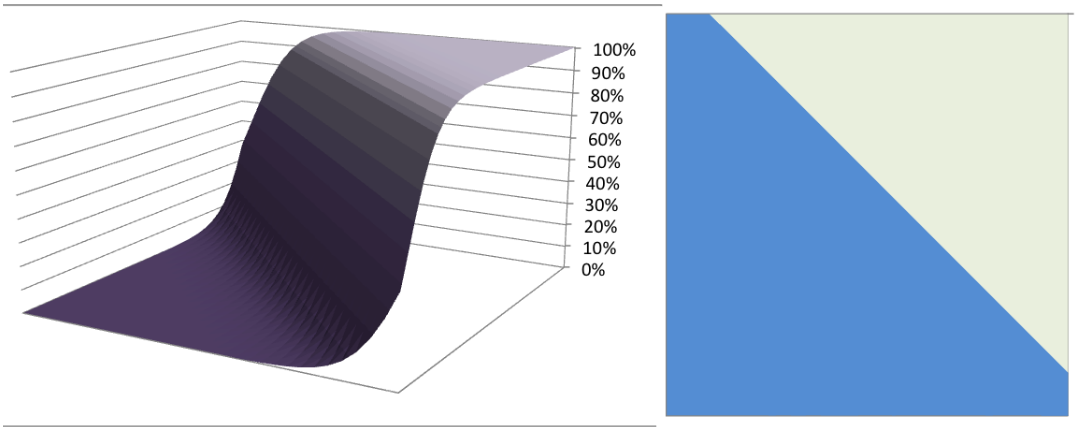


Figure 4: A two dimensional logistic curve and the resulting decision boundary when the decision coefficient is set to 0.5. [31]

be equal to perfect classification. [1]

4 Smoothing & Interpolation

This section introduces various techniques for smoothing and interpolation and specifically compares the techniques used in Generalized Additive Models to Penalized Splines. It also introduces kriging for spatial interpolation and compares this approach to Gaussian Processes. Finally, the use of Latent Gaussian Processes and their Power Spectral Density representation in order to perform GP regression and reduce computational cost is discussed.

4.1 Splines

In most scenarios the relationships between covariates and outcome are going to be more complex than a simple, linear fit. The simplest approach to tackle nonlinear relationships is to add polynomial terms to the design matrix. While the results of adding polynomial terms of growing order to the linear predictor may be simple to interpret, this approach is both prone to not describing complex relationships very well when the degree of added polynomials is low, and overfitting, should the degree of added polynomials be high. This is visualized based on an example given in [33] in figure 5. Another problem with simple polynomial methods is extrapolation. While high degree polynomials fit the data the model was trained on very well, the out of sample error rate is very high and trying to predict values near or at the limits of the predictor interval amplifies these high prediction errors even more due to the high curvature of higher order polynomials. Also, the higher the order of the polynomial fit, the more coefficients need to be estimated. This can lead to unidentifiability in extreme cases. One way to extend the flexibility of polynomials while avoiding overfitting is the construction of piecewise polynomial fits. Here, the covariate is split up into multiple distinct intervals and a polynomial of low order is then fit to each of the intervals separately. To avoid discontinuities from interval

[1] Technically, an AUC equal to 0 would also correspond to perfect classification, as inverting class labels would yield the desired output in such a case.

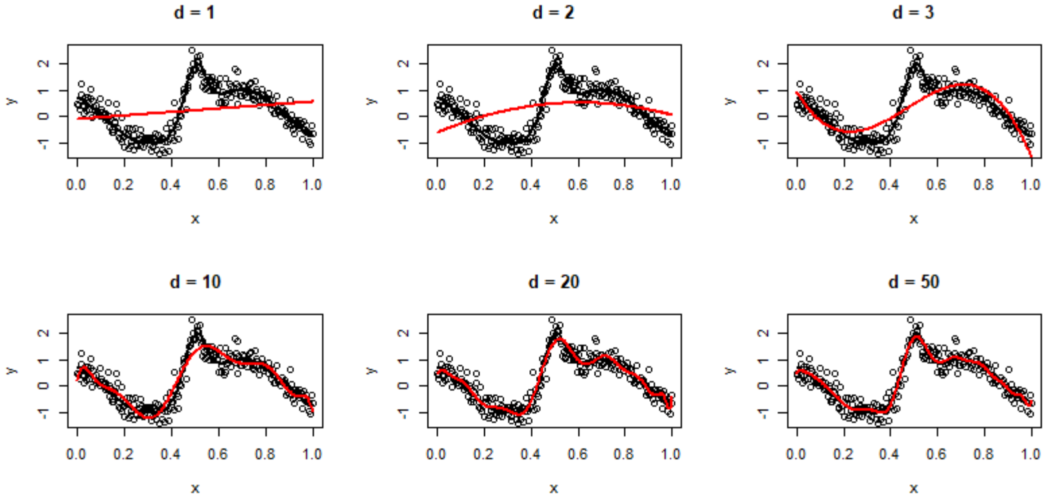


Figure 5: Fitting a polynomial of low degree does not capture the underlying function very well. Similarly, fitting polynomials of a large degree overfits the data and in turn leads to a high error rate for out of sample predictions and very unstable predictions at the boundaries of the interval.

to interval, the resulting function is additionally limited to be at least $(d-1)$ times differentiable for a polynomial of degree d. [33] While this approach generally yields sensible results, both the location and the number of "knots" can have a drastic influence on the fitted curve.

While strategies such as quantile based distribution of knots, or spread of knots based on visual inspection of a scatterplot exist, none of these strategies really answer the question of how many intervals the data should be split up into. [34] A strategy that leads up to Generalized Additive Models, discussed later, is an approach based on regularization that is comparable to to methods like Ridge Regression.

4.2 Penalized Splines

One way of formalizing interval based methods are truncated polynomials. A model of the form

$$y_i = \gamma_1 + \gamma_2 z_i + \ldots + \gamma_{l+1} z_i^l + \gamma_{l+2} \left(z_i - \kappa_2\right)_+^l + \ldots + \gamma_{l+m-1} \left(z_i - \kappa_{m-1}\right)_+^l + \varepsilon_i$$

where

$$(z-\kappa_j)_+^l = \begin{cases} (z-\kappa_j)^l & z \geq \kappa_j \\ 0 & \text{else} \end{cases}$$

consists of a global polynomial of degree l. Additionally, the largest polynomial coefficient may vary in every interval defined by the knots $\kappa_2, \ldots, \kappa_{m-1}$. [35] This makes the model even more flexible for regions of high variability in the data, yet avoids "wasting" degrees of freedom by estimating only a limited number of parameters in regions that are linear. The model is specified by the global polynomial l and the truncated polynomials that capture local structure. Because of this, penalizing high variability in the local polynomials can be achieved through introducing a penalty term to the least squares estimator that incentivizes smaller absolute coefficients of the truncated basis functions. The usual sum of squared differences

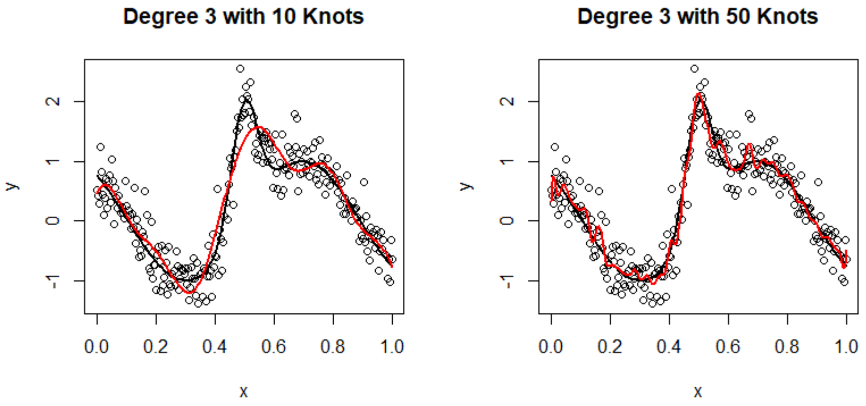


Figure 6: Results of fitting a third degree polynomial to the example data. A low knot count fails to capture the maximum accurately, while a high knot count leads to high curvature of the modeled function, which again causes high out of sample prediction error.

$$\sum_{i=1}^{n} \left(y_i - f\left(z_i\right)\right)^2 = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} \gamma_j B_j\left(z_i\right)\right)^2$$

where the squared differences of the response values y_i to the fitted function $f(z_i)$ can equally be represented as the sum of squared distances of y_i to the products of the coefficients γ_j and the values of their respective basis functions $B_j(z_i)$ is penalized with an additional term.

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} \gamma_j B_j\left(z_i\right)\right)^2 + \lambda \sum_{j=l+2}^{d} \gamma_j^2$$

This is very similar in structure to the penalization in regularization methods such as Ridge Regression. Now, starting off with a sufficient amount of knots to construct the basis functions, the only thing that needs to be estimated, additionally to the model coefficients themselves, is the smoothing parameter λ. Setting λ to 0, the unpenalized sum of squares is recovered. Letting λ tend to ∞, the resulting model would be a line of degree l. This is illustrated in figure 7. With this technique, the position and count of the knots is no longer a priority, as long as there are enough knots available to ensure a satisfactory fit to the data. A common default value is 30 equidistant knots, but quantile based methods are also sometimes used. Simon Wood in [54] also mentions that: “It is possible to start with a fine grid of knots and simply drop knots sequentially, as part of backward selection, but the resulting uneven knot spacing can itself lead to poor model performance. Furthermore, the fit of such regression models tends to depend quite strongly on the locations chosen for the knots.” The penalty λ is usually estimated using cross validation. In matrix notation these penalized models can be written as

$$y = Z\gamma + \varepsilon$$

where γ is a vector of the polynomial coefficients, Z is a design matrix constructed from data and interval, or basis specifications and ε is a vector of independent $N\left(0, \sigma^2\right)$ random variables. The resulting Penalized Least Squares (PLS) criterion

$$\text{PLS}(\lambda) = (\boldsymbol{y} - \boldsymbol{Z\gamma})'(\boldsymbol{y} - \boldsymbol{Z\gamma}) + \lambda\boldsymbol{\gamma}'\boldsymbol{K\gamma}$$

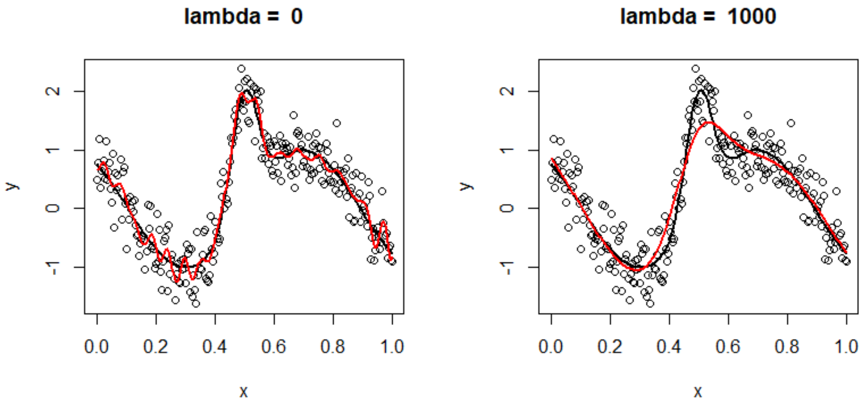


Figure 7: The result of a unpenalized (left) and a penalized (right) model fit. $\lambda = 0$ leads to overfitting, while strong penalization cannot model the maximum in the data adequately.

yields the penalized least squares estimator $\hat{\gamma}$

$$\hat{\gamma} = \left(Z'Z + \lambda K\right)^{-1} Z'y$$

and can be obtained by either solving this system of equations directly, or employing Restricted Maximum Likelihood (REML).

4.3 Generalized Additive Models

A simple way of generalizing univariate smoothing of the form

$$y_i = \alpha + f\left(x_i\right) + \epsilon_i$$

where every response y_i is the sum of an intercept parameter α and a smooth function (x_i) plus an independent error term ϵ_i, is a simple additive structure.

$$y_i = \alpha + f_1\left(x_i\right) + f_2\left(v_i\right) + \epsilon_i$$

Here, the two explanatory variables x and v are each modeled with their own smooth function. It should be noted, that this additive structure is a strong constraint of general bivariate functions $f(x, v)$, and needs to be fit to the data under additional identifiability restrictions, as adding any arbitrary constant to $f(x)$ and subtracting the same constant from $f(v)$ would not influence model predictions [55].

Models of this additive structure were first extended in the same manner the simple linear model is extended to a GLM by Hastie & Tibshirani [51] in 1986 as "Generalized Additive Models". In this context, the linear predictor η predicts a known, smooth and monotonic function of the expected value of the response. [56] Basic additive models can be estimated by solving the penalized least squares equation featured above, for generalized models restricted maximum likelihood methods are needed. Specifically, the popular implementation in the R package "mgcv" [57] uses the Penalized Iterative Least Squares (PIRLS) algorithm. In short, the following steps are repeated until a convergence criterion is reached:

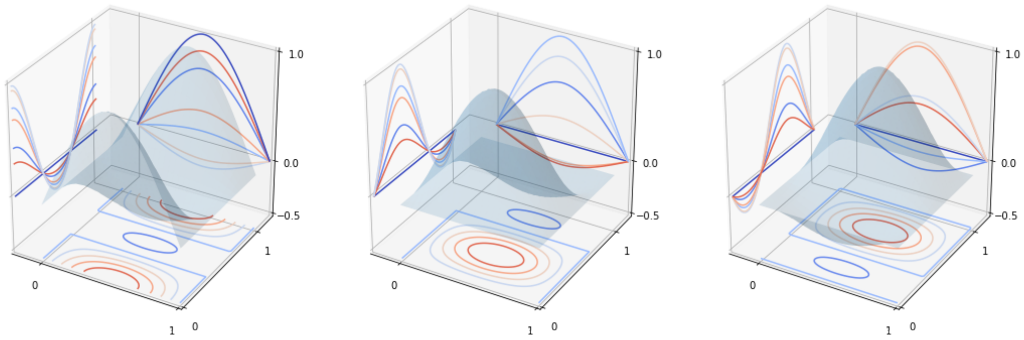


Figure 8: A visual example of Tensor Product bases. The two one dimensional vector spaces (visible on the background) containing a single smooth function for each covariate are "multiplied" together, resulting in a two dimensional surface.

1. Given the current estimate for the linear predictor $\hat{\boldsymbol{\eta}}$ and the corresponding mean response vector $\hat{\boldsymbol{\mu}}$ calculate: $w_i = \frac{1}{V(\hat{\mu}_i)g'(\hat{\mu}_i)^2}$ and $z_i = g'\left(\hat{\mu}_i\right)\left(y_i - \hat{\mu}_i\right) + \hat{\eta}_i$ Where $\operatorname{var}\left(Y_i\right) = V\left(\mu_i\right)\phi$ is a function proportional to the variance of Y_i and g is the chosen link function.

2. Define W as a diagonal matrix of the weights w_i and minimize:

$$\|\sqrt{\mathbf{W}}\mathbf{z} - \sqrt{\mathbf{W}}\mathbf{X}\boldsymbol{\beta})\|^2 + \lambda_1\boldsymbol{\beta}^\top\mathbf{S}_1\boldsymbol{\beta} + \lambda_2\boldsymbol{\beta}^\top\mathbf{S}_2\boldsymbol{\beta}$$

   with respect to $\boldsymbol{\beta}$, then update the estimates for $\hat{\boldsymbol{\eta}}$ through $\hat{\boldsymbol{\eta}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\mu}}$ through $\hat{\mu}_i = g^{-1}\left(\hat{\eta}_i\right)$ [56]

Values at which the optimization is initialized can be defined manually, but are set through a run of linear regression in most cases. Similar to the single variable case, the smoothness terms λ_i for every predictor variable can be estimated with Generalized Cross Validation or Un-Biased Risk Estimators as introduced in 1979 by Craven and Wahba [40]. Automatic estimation of the smoothness parameters may fail when there are little data available, more so, in cases when there are multiple smooth terms present in the model. In some cases the optimization process may also struggle with occasional local minima produced by GCV, but this can usually be avoided by supplying sensible starting values to the optimization routines.
The implementation of GAMs in "mgcv" allows for flexible model specification and extends these ideas for smooth function estimation to include interaction terms with basis expansions based on thin plate splines or tensor product bases as depicted in figure 8.

Another useful property of GAMs is the possibility of including yet another penalization term in the smooth terms themselves which can be used for model selection. Especially for sparse data it may be unclear if a smooth term of a predictor variable should be included in the model specification or not. "Smooth classes cs.smooth and tprs.smooth (specified by "cs" and "ts" respectively) have smoothness penalties which include a small shrinkage component, so that for large enough smoothing parameters the smooth becomes identically zero. This allows automatic smoothing parameter selection methods to effectively remove the term from the model altogether. The shrinkage component of the penalty is set at a level that usually offers negligible contribution to the penalization of the model, only becoming

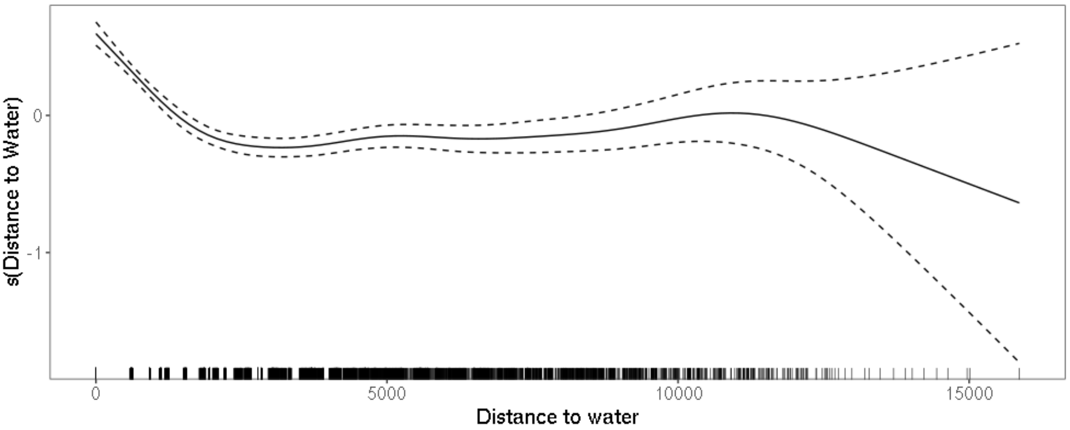


Figure 9: The effect of the distance to the nearest large body of water on the linear predictor. Values smaller than about 1000 meters tended to have a positive effect on the odds of finding a site at a given location, for larger distances to the nearest body of water the estimated effect on the odds is negative. The dashed lines indicate estimated confidence intervals for the smooth function. The marks on the bottom of the graph correspond to the values present in the data set. At about 9000 meters the lack of available data causes the confidence intervals to gradually become broader and overlap 0. For illustration purposes the model did not include any other variables.

effective when the term is effectively 'completely smooth' according to the conventional penalty." [52]

An example of a smooth function fit to the data of Bavarian iron age settlements is depicted in figure 9.

The last couple of sections have introduced different methods for the representation and estimation of models based on smooth functions. A theme that unites all of these smoothing approaches is the introduction of penalization terms similar to penalty terms used in regularization. While adding penalization terms to the estimators of models seems arbitrary at first, they can be justified from a Bayesian point of view. Perhaps the most popular example for this is Ridge Regression (Tikhonov Regularization). Here the penalized sum of squares

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(y - X\beta)^T(y - X\beta) + \lambda\|\beta\|_2^2$$

needs to be minimized. Under the assumption that the design matrix X is fixed, the OLS model presupposes that the conditional distribution of the response y is

$$y|X, \beta \sim \mathcal{N}\left(X\beta, \sigma^2 I\right)$$

Now, setting a prior for β as independent normal random variables with a constant variance τ i.e. $\beta \sim \mathcal{N}\left(0, \tau^2 I\right)$ allows deriving the posterior distribution of β:

$$\begin{aligned} p(\beta|y, X) &\propto p(\beta) \cdot p(y|X, \beta) \\ &\propto \exp\left[-\frac{1}{2}(\beta - 0)^T \frac{1}{\tau^2} I(\beta - 0)\right] \cdot \exp\left[-\frac{1}{2}(y - X\beta)^T \frac{1}{\sigma^2}(y - X\beta)\right] \\ &= \exp\left[-\frac{1}{2\sigma^2}(y - X\beta)^T(y - X\beta) - \frac{1}{2\tau^2}\|\beta\|_2^2\right] \end{aligned}$$

From this, the Maximum a-Posteriori estimate can be computed:

$$\begin{aligned}\hat{\beta} &= \underset{\beta}{\operatorname{argmax}} \exp\left[-\frac{1}{2\sigma^2}(y - X\beta)^T(y - X\beta) - \frac{1}{2\tau^2}\|\beta\|_2^2\right] \\ &= \underset{\beta}{\operatorname{argmin}} \frac{1}{\sigma^2}(y - X\beta)^T(y - X\beta) + \frac{1}{\tau^2}\|\beta\|_2^2 \\ &= \underset{\beta}{\operatorname{argmin}} (y - X\beta)^T(y - X\beta) + \frac{\sigma^2}{\tau^2}\|\beta\|_2^2\end{aligned}$$

Which for $\lambda = \frac{\sigma^2}{\tau^2}$ is equal to the ridge regression estimator. [49]

4.4 Geostatistic Data & Kriging

According to Toblers first law of geography, “Everything is related to everything else, but near things are more related than distant things.” [50] This is true for interpolation of (reasonably well behaved) functions, but in the case of geographic data the intrinsic relations between measurements can be leveraged for more accurate prediction.

In a geostatistical context, measurement values m can be split up into location data s (coordinates) and additional information data z. (The theory presented in this section is largely based on [27])

Usually, measurements have been made at certain locations and predictions at different locations are to be made. It is helpful to view the measurements m as realizations of a stochastic process $Z(s_i)$. Similar to other methods of smoothing, a weighted estimator may be defined in the following way:

$$\hat{Z}(s_0) = \textstyle\sum_{i=1}^{n} w_i Z(s_i)$$

But here, this sum is not penalized based on a smoothness parameter λ or the degree of polynomials. The weights w_i are based on the distance between the known locations s_i and new location s_0. This technique is commonly known as “Kriging” and was developed by Danie Krige in 1951 to predict ore deposits based on spatially dependent measurements.

To develop a method for parametric estimation of Kriging weights, additional asuptions about the structure of the data generating stochastic process are needed. The simple model:

$$Z(s) = \mu(s) + \epsilon(s)$$

splits the process Z into a mean function μ and errorterms ε. Assuming that every realization $Z(s_i)$ has been drawn from the same distribution, i.e. belongs to the set of possible locations for settlements in the iron age, is known as “stationarity” of a stochastic process. While stationarity implies complete equivalence of the distributions, the first two moments are of particular interest in geostatistics and thus, “weak stationarity” is defined as follows:

If the expected value $E[Z(s)]$ is equal to a constant value μ and the covariance of two different locations $\operatorname{Cov}[Z(s + \boldsymbol{h}), Z(s)]$ is a function of only the distance $\boldsymbol{h}$ between the points $c(\boldsymbol{h})$, then the stochastic process Z is weakly stationary.

If additionally, the covariance does not depend on the direction of the distance vector $\boldsymbol{h}$, the process is called isotropic.

Anisotropic processes can be modeled by introducing a rotation term to the distance calculations or through splitting of the process of interest into multiple sub processes. Details for this are given in [18]. In any case, estimating covariance matrices for parametric Kriging requires an additional structural assumption, because predicting out of sample values may require covariance values for distances not present in the training set. The covariance structures considered for the Iron Age data are:

The spherical covariance:

$$\gamma(h) = \begin{cases} 0 & \text{for } h = 0 \\ c_0 + c_1 \left(\frac{3h}{2a} - \frac{1}{2} \left(\frac{h}{a} \right)^3 \right) & \text{for } 0 < h \leq a \\ c_0 + c_1 & \text{for } h \geq a \end{cases}$$

The exponential covariance:

$$\gamma(h) = \begin{cases} 0 & \text{for } h = 0 \\ c_0 + c_1 \left(1 - e^{-h/a} \right) & \text{for } h \neq 0 \end{cases}$$

And the Matérn class of covariance functions:

$$\gamma(h) = \begin{cases} 0 & \text{for } h = 0 \\ c_0 + c_1 \left[1 - \frac{1}{2^{\kappa-1}\Gamma(\kappa)} \left(\frac{h}{a} \right)^\kappa K_\kappa \left(\frac{h}{a} \right) \right] & \text{for } h \neq 0 \end{cases}$$

Here K_κ is equal to a modified Bessel function of the second kind. For κ equal to 1/2 the exponential covariance function is obtained. The Matérn class are a generalization of the gaussian covariance function. They are differentiable only finitely many times and according to Stein [48] more realistically describe physical processes because of this.

The main reason why Kriging is so attractive for making predictions, is that it minimizes the mean squared prediction error (MSPE).

$$\text{MSPE} = E\left[\left(Z(s_0) - \hat{Z}(s_0) \right)^2 \right] = \text{Var}\left[Z(s_0) - \hat{Z}(s_0) \right]$$

Ordinary Kriging assumes a constant, but unknown mean μ.

$$Z(\boldsymbol{s}) = \boldsymbol{\mu} + \delta(\boldsymbol{s})$$

Where $\delta(.)$ is a zero-mean stationary process.[27] This can of course be extended through representation of μ as a linear combination of smooth functions. This is known as “universal Kriging”. A detailed explanation is given in [39]. This is the method that was used to model the spatial relationship for the Bavarian Iron Age data, as it uses the coordinates of sites and non-sites to model the spatial relationship in the data. This is especially useful, since according to [18] it can be shown that universal Kriging for geographic data is equivalent to a two dimensional penalized smoothing problem where the covariance assumption serves as a “smoothness penalty”.

A way of unifying numerous approaches to smoothing and regularization are Gaussian processes. It can be shown, that Gaussian processes are mathematically equivalent to many well known models, including Bayesian linear models, spline models, and even large neural networks (under suitable conditions), and are closely related to others, such as support vector machines. [10]

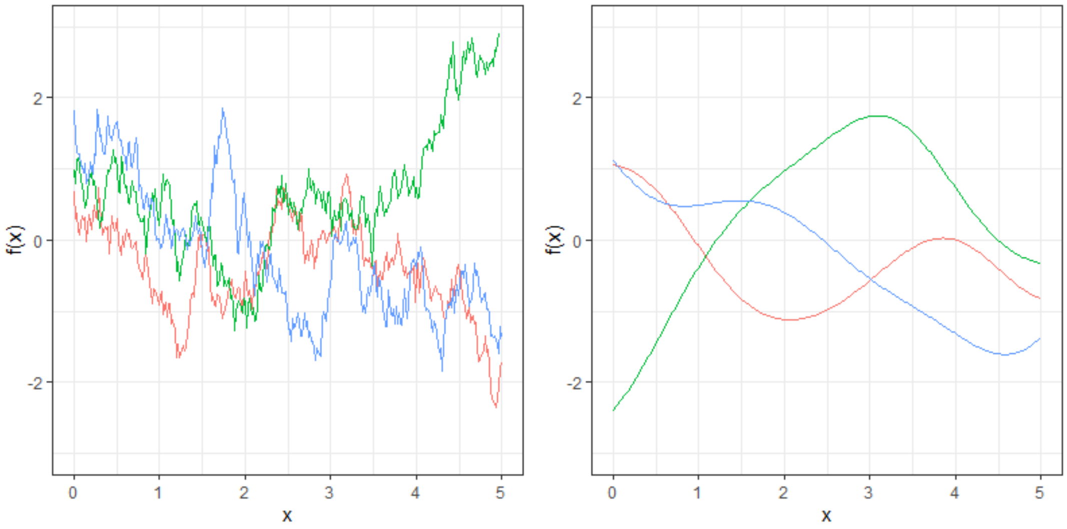


Figure 10: Three function samples from the function spaces defined by the Matérn $\nu = 0.5$ (left) Matérn $\nu = 2.5$ covariance kernels. The higher ν parameter favours smoother functions, while a smaller value assigns higher probabilities to functions that are coarse.

4.5 Gaussian Processes

From a Bayesian point of view, the smoothing methods introduced in the previous sections can be described as defining a "prior" for the class of functions that will be fit to the model i. e. restricting the possible functions to be polynomials of degree 1 (in the case of the linear model). While penalized smoothers are already very powerful tools, it is easy to see that they are limited. For example, modeling stock market time series data would, in all but the simplest cases, require a large amount of parameters to be estimated and would still not be suited for prediction of future prices and risk assessment.

Evidently, a better approach would be not to restrict the class of function itself, but rather assign a prior probability to every possible function. Here prior knowledge is incorporated into the model indirectly, by assigning functions deemed to likely describe the data a higher probability. And while the space of all possible functions in any given interval is infinite, the properties of Gaussian Processes allow for computation in finite time.

A Gaussian Process is an infinite dimensional generalization of the normal distribution. Where a distribution describes samples of random variables (or vectors of random variables in the multivariate case) a stochastic process governs the properties of functions.

Visual examples of such "priors over functions" are given in figure 10. Mathematically Gaussian processes are defined as probability distributions over infinite-dimensional Hilbert Spaces of functions [6] They are specified by a mean and a covariance in much the same way a finite-dimensional Gaussian distribution may be specified.

$$f \sim \mathcal{GP}(m, k)$$

As with finite-dimensional Gaussian distributions, Gaussian processes are specified with a mean and covariance, but both mean and covariance are now defined as

functions instead of fixed values. For convenience the mean function is defined to be 0, but even if information about the average behavior of the modeled function is available, it is always possible to separate the mean function from the Gaussian process by linearity. Because of this, Gaussian processes can be fully specified by their covariance function alone.
The covariance function, or covariance kernel, $k\left(x_1, x_2\right)$ controls how the realizations of a Gaussian process vary around the mean function. [7] The most broadly known kernel is the "exponentiated quadratic kernel." (RBF Kernel)

$$k\left(x_1, x_2\right)=\alpha^2 \exp \left(-\frac{\left(x_1-x_2\right)^2}{\rho^2}\right)$$

It has a few very useful mathematic properties which caused it to become a staple for both GPs and Support Vector Machines. For one, it is a universal kernel, which for a continuous kernel implies that it may approximate any arbitrary continuous target function uniformly on any compact subset of the input space. [12] It is also differentiable infinitely many times which, according to Stein [48] may yield unrealistic results for physical processes, as the observation of only a small continuous subset of space should yield the entire underlying function. Stein also proposed the Matérn class of covariance functions as a generalization of the exponentiated quadratic kernel to tackle this problem. The lengthscale parameter ρ determines the "wiggliness" of the functions, comparable to the frequency of a wave. The variance parameter α determines the average distance of the function away from its mean, similar to the amplitude of a wave. [16] Another key attribute of kernel functions is the ability to add or multiply different kernel functions together which in turn yields another kernel function. This is particularly useful for modeling the combined effects of different variables on the same outcome, as when two one dimensional kernels are multiplied with each other the result is a prior over functions that maps from the combined two dimensional predictor space to the one dimensional output space. Another side effect of this, is that non-isotropic GPs can be modeled by simply choosing different GP priors for two different predictors. These product kernels have the form:

$$k_{\text{product }}\left(x, y, x^{\prime}, y^{\prime}\right)=k_x\left(x, x^{\prime}\right) k_y\left(y, y^{\prime}\right)$$

To actually perform statistical inference with GPs, their "marginalization property" is employed. Since data sets in practice are always limited to a finite number of observations, the resulting GP will always marginalize to a multivariate Gaussian distribution with a mean $\mu_i=m\left(x_i\right)$ and covariance matrix $\Sigma_{ij}=k\left(x_i, x_j\right)$. Therefore any inference on GP models boils down to working with multivariate Gaussian distributions. Considering, for example, that univariate smoothing can be formalized like

$$f(y|x, \sigma)=\mathcal{N}(y|f(x), \sigma)$$

where a GP prior is assumed for the function $f(x)$. After making measurements the prior belief is updated with the information in the data (more on that in section 5) and a GP posterior can be recovered. Under the assumption of Gaussian measurement errors the resulting posterior is also a GP, with analytic mean and covariance functions. [7] The posterior GP keeps the mean function of the prior GP, but the resulting covariance function is

$$k^{\prime}\left(x_1, x_2\right)=k\left(x_1, x_2\right)+\sigma^2 \delta\left(x_1-x_2\right)$$

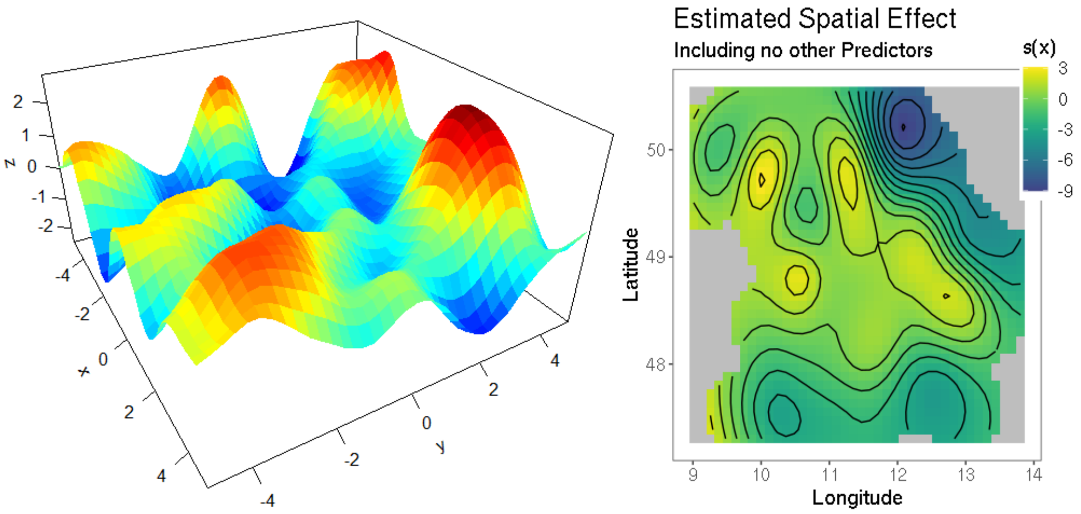


Figure 11: Left side: The result of multiplying two one dimensional RBF kernels together. The original kernels were only dependent on x and y individually, sampling from the combined kernel returns surfaces similar to the one depicted. Right side: Combining two RBF kernels to estimate the spatial effect on the outcome of finding settlement traces. The darker regions in the south and north east have negative effects, the yellow and bright green regions near the center have positive effects. For visualization purposes only the longitude and latitude were included in this model.

Given measurements $\{y, x\}$ a joint multivariate Gaussian distribution over all covariates can then be estimated (or in this special case analytically constructed) and predictions for new values x' can be made by sampling from the posterior and averaging over the results.
In the case of the archaeological settlement data, many of the covariates are not Gaussian, so in practice, the latent multivariate Gaussian representation of GPs is used to approximately recover the posterior GP through sampling. A visual example for this is given in figure 12.

4.6 Gaussian Process Regression

For the prediction of real valued outputs y' based on new data x' a GP prior over functions can be used to specify the distribution outcomes $\boldsymbol{y} = (f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_n))$ so the probability of observing particular values y' given observed values x is given by:

$$\begin{aligned}
P(y'|x) &= \int P(y', y|x)\,dy \\
&= \frac{1}{P(x)} \int P(y'|y)\, P(y)P(x|y)dy \\
&= \int P(y'|y)\, P(y|x)dy
\end{aligned}$$

where $P(y', \boldsymbol{y})$ is defined as the joint distribution of y', y. [13] So the predictive distribution for y in the case of a regression problem with Gaussian noise can be found by marginalization. The prediction problem for site and non-site data requires a link function to limit prediction outcomes to the $[0; 1]$ interval. “For binary

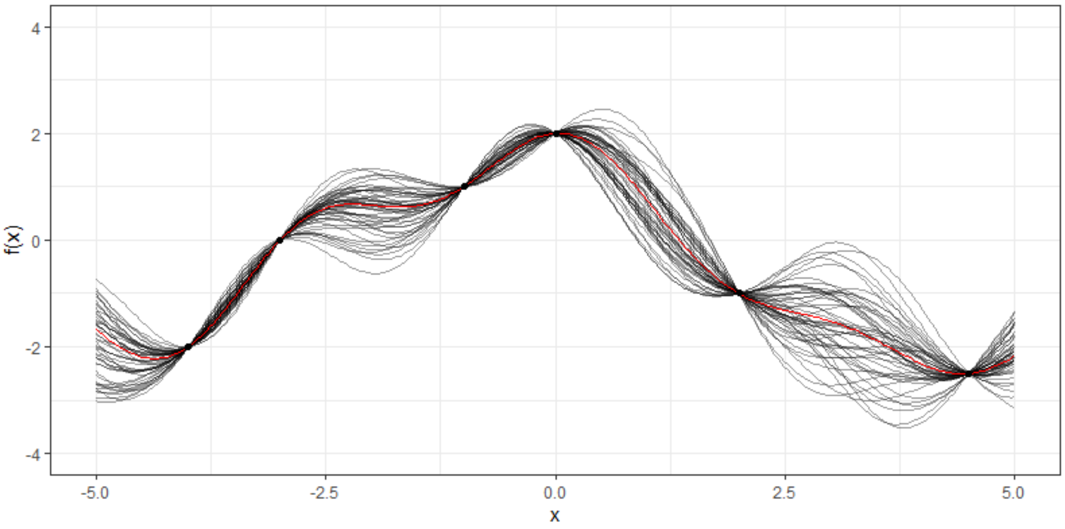


Figure 12: GP smoothing in practice. A RBF kernel is combined with information from the sampled points. 30 functions from the resulting posterior GP were drawn and averaged. The mean function is colored red.

classification the basic idea behind GP prediction is very simple – we place a GP prior over the "latent function" $f(x)$ and then "squash" this through the logistic function to obtain a prior on $\pi(\mathbf{x}) \triangleq p(y = +1|\mathbf{x}) = \sigma(f(\mathbf{x}))$ [11] The function f is referred to as "latent", because no values of it are directly observed. This detour over a latent function can be imagined as a projection of the input values to this latent space. The transformed input variables are then used to fit the final logistic model. The resulting predictive distribution can no longer be recovered analytically, but Markov Chain Monte Carlo (MCMC) methods can be used to approximate this distribution computationally.

A challenge of performing GP regression is their large computational complexity. The need to calculate $n \times n$ covariance matrices where n is the number of available data points causes GP algorithms, in theory, to be of order $\mathcal{O}(n^3)$. With smaller data sets and modern computing power, this is not really an issue, but with a data set containing thousands of points, GP calculations tend to become intractable – especially in the multivariate case. Calculating the exact GP solutions for the Bayesian model of the 12222 points in the Bavarian settlement data set would have taken upwards of two weeks. There are many different ways of approximating GPs, including subsetting of data and low-rank matrix approximations. The "brms" [9] package that was used as an interface from R to Stan in this thesis implements a novel method proposed by Solin and Särkkä in 2014 that, theoretically, reduces the computational complexity to $\mathcal{O}(nm^2)$ with n data points and m specified basis functions. In short, this method falls in the class of low-rank matrix approximations, in that it approximates the covariance matrix of the GP through simple functions of the spectral density of the GP. [1] Spectral analysis splits the "signal" of a stochastic process into a sum of periodic functions each representing a part of the total variability in the process. Similar to how analog radio signals can be split up into carrier waves and signal waves. A detailed proof of this is given in [1]

5 Bayesian Methods

This section briefly discusses central concepts in Bayesian Inference and builds upon these ideas to explain the No-U-Turn-Sampler developed by Hoffman & Gelman [36] and its use in Hamiltonian Monte Carlo.

5.1 Markov Chain Monte Carlo

The ultimate undertaking in statistical computing is evaluating expectations with respect to some distinguished target probability distribution. [2] According to Bayes' theorem, this can be represented in the form

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}$$

where $P(\theta|x)$ is the posterior probability distribution of the parameter θ given the data sample x, $P(x|\theta)$ corresponds to the likelihood function of the data x given the parameter θ and $P(\theta)$ represents the prior probability distribution assigned to θ. To recover the posterior distribution of θ from the numerator, it is devided by the marginal probability $P(x)$, this can be imagined as a normalizing constant that ensures that the result is a proper probability distribution. In cases where the prior and posterior distributions are conjugate, this can be solved analytically. Otherwise, according to the law of total probability,

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{\int_{-\infty}^{\infty} P(x|\theta)P(\theta)d\theta}$$

the calculation of the marginal probability $P(x)$ involves solving high dimensional integrals which is not computationally feasible in all but the simplest cases. Fortunately, as long as it is possible to obtain a function that is proportional to the target distribution,

$$P(\theta|x) \propto L(x|\theta)P(\theta)$$

independent samples can be generated from the posterior distribution directly. Using computer generated random draws to simulate physical systems is known as Monte Carlo methods. They were originally designed in the 1930s to aid in studying the effects of neutron diffusion.

An example for generating samples from a target distribution in the univariate case is adaptive rejection sampling. In order to generate samples from a target density $\pi(x) = f(x)/K$ where $f(x)$ is an unnormalized density function and K is a normalizing constant, first a density $h(x)$ must be found from which random draws can be simulated. If there exists a known constant c such that $f(x) \leq$ ch(x) then random draws from $\pi(.)$ can be obtained in the following way: [45]

Algorithm 1: Adaptive Rejection Sampling

```
Generate a candidate $Z$ from $h(.)$ and a value $u$ of the uniform distribution
  on $(0,1)$
if $u \leq f(Z)/ch(Z)$ then
    return $Z = y$;
else
    Start over with a new candidate $Z$ from $h(.)$;
end
```

Even when choosing c carefully, this method may result in a very large value of rejected samples. This effect is amplified when trying to sample from distributions of higher dimensionality. As the dimensionality of the target distribution increases, the ratio of acceptance space to total sampling space shrinks drastically. More efficient methods are needed.

One of the most popular Monte Carlo sampling algorithms is Random Walk Metropolis. While it is generally impossible to infer the global structure of the posterior distribution without calculating the normalization constant, it is still possible to locally explore posterior space by generating *dependent* samples. To reiterate, the main goal here is to generate samples at points in parameter space whose frequency varies in proportion to the corresponding values of the posterior density. [29] Similar to accept/reject sampling, a random walk through the posterior space can be constructed where a decision is made at every step, if the generated point should be included in the generated sample or not. If every single point is accepted, this method is equal to sampling uniformly from the entire parameter space. To make use of the "height" information available in the unnormalized posterior this random stepping routine can be extended to only accept proposed steps if the function value of the unnormalized posterior at the proposed height is larger than the one at the current point. Clearly, this algorithm is also flawed, as it will only ever accept points leading towards the maximum of the unnormalized posterior. It can be proven, that always accepting proposed steps θ_{t+1} if $p(\theta_{t+1}|x) \geq p(\theta_t|x)$ and accepting proposed steps with the probability $\frac{p(\theta_{t+1}|x)}{p(\theta_t|x)}$ if $p(\theta_{t+1}|x) < p(\theta_t|x)$ leads to the desired behaviour. If additionally the proposed points are generated based on a multivariate Gaussian distribution centered on the current point, the entire procedure generates a Markov chain whose stationary distribution is the desired posterior distribution. Random walk Metropolis works for an arbitrary number of dimensions in theory, and, in some cases, the structure of the posterior can be exploited with techniques like "Gibbs Sampling", to allow for faster exploration of the posteror space. But the random walk behaviour of RWM is still problematic for distributions that have long tails. Here the probabilistic nature of the accept/reject rule can cause the random walk to take a very long time to sufficiently explore posterior space. While these effects can be compensated for by parameter transformation, this method still struggles in cases where the posterior distribution is multimodal or variables exhibit high correlation, as shown in figure 13

5.2 The Nuts and Bolts of Hamiltonian Monte Carlo

"In high-dimensional parameter spaces probability mass $\pi(x)dx$, and hence the dominant contributions to expectations, concentrates in a neighborhood called the typical set. In order to accurately estimate expectations we have to be able to identify where the typical set lies in parameter space, so that we can focus our computational resources where they are most effective." [3]

While the Markov chains generated by RWM always converge to the stationary distribution given infinite time, in real world scenarios this can be problematic. Stepping through the posterior space with arbitrarily small steps can increase the acceptance rate of proposed locations, but by doing this the chain will take an even longer time to fully explore the posterior space. To make more informed guesses on where to go next, additional information about the posterior needs to be ex-

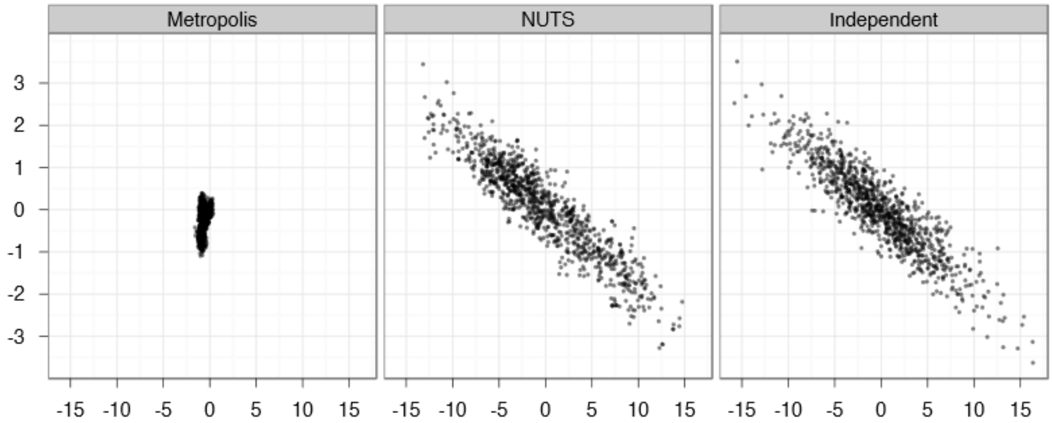


Figure 13: Samples generated by random-walk Metropolis and NUTS. The plots compare 1000 independent draws from a highly correlated 250-dimensional distribution (right) with 1000000 samples (thinned to 1000 samples for display) generated by RWM (left), and 1000 samples generated by NUTS (middle). Only the first two dimensions are shown here.[37]

ploited. A very elegant way of accomplishing this is utilizing the geometry of the unnormalized posterior itself. It can be useful to imagine the surface of the posterior distribution as a mountain. RWM only used the "height" (outcomes of evaluating the unnormalized posterior at the current and proposed locations) information. By calculating derivatives of the posterior at the current location, information about the local geometry can be extracted. Hamiltonian Monte Carlo (HMC) functions in a very similar way and, as a result, generates efficient paths through the posterior landscape.

HMC constructs a vector field aligned with the typical set of the target distribution. So when following this vector field for a set amount of steps, all points on the path to the final location are members of the typical set by construction. Constructing this vector field only based on the gradient of the target density is not sufficient. Because the gradient depends on the parametrization of the target density, the gradient will be steepest in regions around the mode of the density which are not guaranteed to be aligned with the typical set. To rectify this misalignment, HMC simulates a physical system. As described by Betancourt in [3] it is perhaps more intuitive to imagine a satellite orbiting a planet instead of a path through a parameter space. The planet would be equivalent to the mode of the target distribution, the satellite corresponds to the current location in parameter space, and the gravitational field of the planet matches the gradient of the target distribution. To "place the satellite in orbit" i.e. find a path along the typical set, it needs to be given auxiliary momentum that is both large enough to avoid falling along the gravitational field, and small enough to avoid drifting off into space. In other words, walking along the typical set requires a probabilistic structure that guarantees conservation of momentum. In physics, these conservative mechanical systems are described using Hamiltonian Mechanics. In a Hamiltonian context, physical systems are described by coordinate pairs $r = (\boldsymbol{q}, \boldsymbol{p})$ where $\boldsymbol{q}$ is the set of coordinates from the original reference frame and $\boldsymbol{p}$ are a set of auxiliary coordinates describing momentum. This new space of 2D dimensions is called the phase space of the system. The phase space representation ensures invariance to reparameterization. Equally, the target distribution is represented in

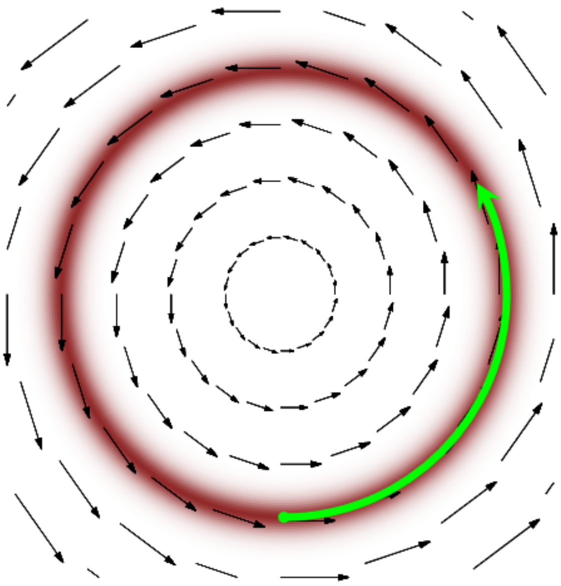


Figure 14: A vector field assigns a vector to every point in the parameter space. When those directions are aligned with the typical set (red) we can follow them like guide posts, generating coherent exploration of the largest distribution. [4]

phase space as the “canonical distribution”. The canonical distribution is defined as a conditional distribution over the auxiliary momentum, which ensures that if the momentum is marginalized out, the target distribution is recovered. [4] Because the canonical density $\pi(q, p)$ is by construction invariant to parameterization it can be defined through a Hamiltonian function in much the same way as conservative physical systems.

$$\pi(q, p) = e^{-H(q,p)}$$

And because the canonical distribution has been defined as a conditional distribution, the Hamiltionian can be broken up into two parts.

$$\begin{aligned} H(q, p) &= -\log \pi(p|q) - \log \pi(q) \\ &\equiv K(p, q) + V(q) \end{aligned}$$

Where $K(p, q)$ is a function of the location variables q and auxiliary momentum variables p and $V(q)$ is a function of the location variables. This can be thought of as terms describing kinetic and potential energy respectively. The potential energy is solely defined by the target distribution, the kinetic energy must be constrained to avoid both “crashing” to the mode of the canonical distribution and “drifting off” into regions of low interest. Now, a vector field that is aligned with the typical set of the canonical distribution can be obtained through applying Hamilton’s equations: [5]

$$\begin{aligned} \frac{\mathrm{d}q}{\mathrm{d}t} &= +\frac{\partial H}{\partial p} = \frac{\partial K}{\partial p} \\ \frac{\mathrm{d}p}{\mathrm{d}t} &= -\frac{\partial H}{\partial q} = -\frac{\partial K}{\partial q} - \frac{\partial V}{\partial q} \end{aligned} \tag{1}$$

Where the resulting vectors only depend on the partial derivatives of the kinetic energy and the negative gradient of the log-posterior. This recovers the physical analogy of the gravity well. Because taking the log is a monotonous transformation, the argument can be made that the “mountains” of high density regions in the posterior directly correspond to the “valleys” obtained by looking at the negative log-posterior. Following all this, an efficient Markov transition can be generated by

1. Sampling an initial point from the canonical distribution $p \sim \pi(p|q)$
2. Sampling a value for the momentum with which the transition is made.
3. Taking discrete steps along the vector field i.e. solving Hamilton’s equation using numeric integration.

The generated paths are going to move through the typical set of the target distribution very quickly, but both integration time (the number of discrete steps taken for every path) and step size must be chosen in a way that ensures no additional waste of computation time. Even with modern automatic differentiation techniques, calculating high dimensional gradients is still computationally expensive. Even more so, considering that small step sizes may lead to a higher needed step count per transition. And just increasing the step size will not work in most cases as even the best numeric differential equation solvers are prone to inaccuracies when steps are large.

In practice, the first step to solve these issues is to optimize the choice of kinetic energy. A detailed description of this is given in section 4.2 of [2]. What separates the version of HMC implemented in Stan [46] from ordinary HMC is a remarkable solution to both the integration time and step size problems.

5.3 The No-U-Turn Sampler

Non-Hamiltonian Samplers need to be carefully tuned to find appropriate values for the distances between the current position and the proposed position, and to compensate for autocorrelation that is usually present in the resulting chains. “Thinning” is a way of dealing with the latter problem. Here only every n-th sample is kept in the final chain, which should drastically decrease autocorrelation. Because of the way transitions are constructed in HMC, thinning is not needed as long as the target density does not contain regions of extremely high curvature. Nonetheless, thinning may still be employed to decrease memory requirements. But the problem of finding optimal values for the length of every transition remains. It should be mentioned, that numerical inaccuracies that arise when solving differential equations for lots of consecutive steps can be limited by using symplectic integrators. Stan uses the Leapfrog algorithm, which is a special case of the Verlet method that is invariant to the direction of time. [36] [44] This is important, as time invariance is needed to guarantee detailed balance (time reversibility) of the Markov chains. The solution to finding an appropriate point at which to end the path simulation, that is proposed by Hoffman and Gelman in [36], is based on a criterion that measures if a path has made a U-turn. The U-turn criterion makes intuitive sense, as when the simulated path starts turning around and closing back in on the starting position, the distance from the start to the current location no longer increases. This can be interpreted as wasting computation on steps through regions of the typical set that are getting closer to the starting location. Since the goal is to explore the typical set as efficiently as possible, such U-turns should be avoided.

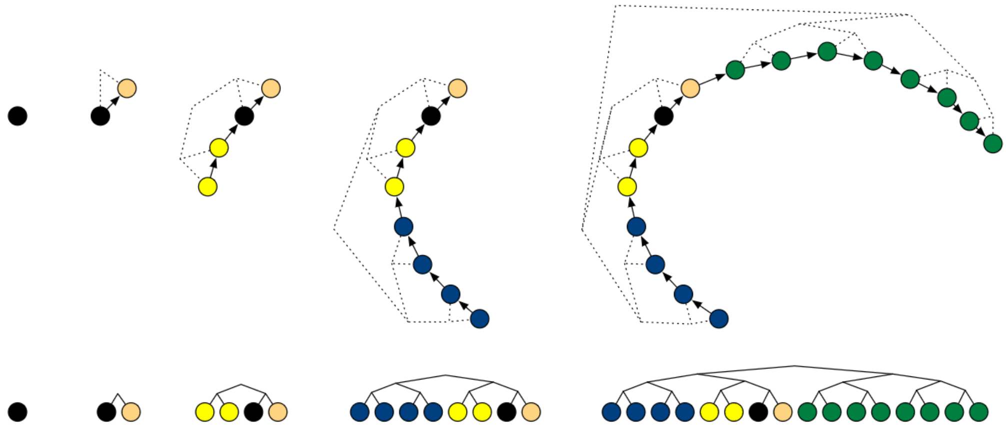


Figure 15: Example of a binary path tree in 2 dimensions. Each doubling chooses an initial direction (forwards or backwards in time) then simulates a path for 2^j leapfrog steps in that direction, where j is the height of the binary tree. The top row shows an example path with chosen directions: Forward, backward, backward, and forward. The bottom row shows the corresponding binary trees. [38]

In short, the dot product between the current momentum of the simulated particle (or the satellite moving around the planet) and the vector from the initial position to the current position is calculated. For every proposed step, the resulting value is proportional to the distance that is gained from the starting point. So once the obtained value is negative the path simulation can be stopped for the current transition. To guarantee time reversibility of the resulting chains, the No-U-Turn-Sampler (NUTS) builds paths not step by step, but by construction of a binary tree for every path. [38] In efficient NUTS, the doubling procedure continues until either a predefined maximum treedepth is reached, or any of the subtrees start a U-turn back to their respective starting locations. After the path simulation halts, the next starting location is sampled from all the points contained in the path. An equivalent to a Metropolis accept/reject step is then undertaken by calculating average acceptance probabilities for all points along the path.
The final problem of choosing an optimal step length to ensure both efficient exploration of the typical set, and avoiding high rejection rates, is then solved by adaptively choosing the step length to warrant a minimum average acceptance rate for the points explored in the final doubling iteration. In most cases, NUTS needs a warmup period to reach the typical set. This is comparable to the "burn-in" period a Markov chain may need to converge to it's stationary distribution. NUTS uses a dual averaging scheme that is outlined in detail on page 17 of [36]
The most popular implementation of HMC using NUTS is the probabilistic programming language Stan. It is written in C++ and can be used natively through the command line or through various interfaces to R or Python. Before discussing the modeling procedure however, a couple of ways to diagnose model fit and sampling procedure should be mentioned.

5.4 Diagnostics

The effective sample size N_{eff} is the estimated size of the *independent* sample contained in the total *dependent* sample that was drawn from the target distribution.

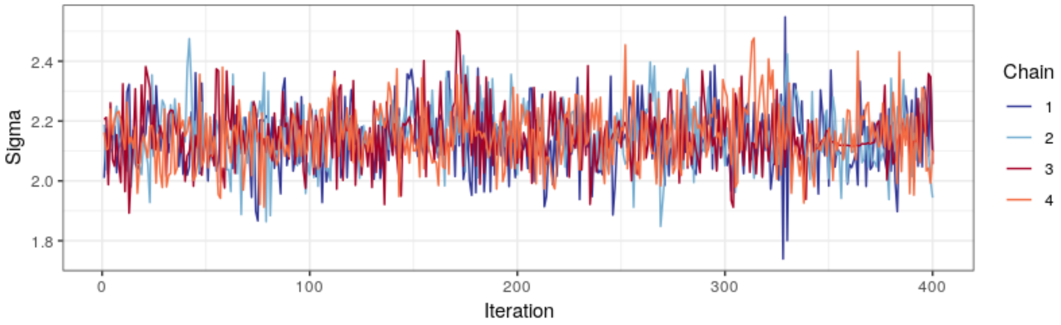


Figure 16: Plotting all chains can help understanding the mixing behaviour of them. None of the chains above seem to stray far from the others, indicating sufficient mixing.

This measure is calculated based on estimating both intra chain autocorrelation and inter chain variance.

$$\hat{\rho}_t = 1 - \frac{W - \frac{1}{M}\sum_{m=1}^{M}\hat{\rho}_{t,m}}{\widehat{\text{var}}^+}$$

The total autocorrelation estimate $\hat{\boldsymbol{\rho}}_{t,m}$ at lag t from chains $m \in (1, \ldots, M)$ are combined with within-sample variance W and multi-chain variance estimate $\widehat{\text{var}}^+$ [47] In an ideal scenario, the autocorrelation of each chain is low and the chains have mixed well, resulting in an effective sample size estimate that is close to the total sample size. Only looking at the effective sample size is not helpful unless it can also be assumed that all chains converged to the target distribution. Because of this, Stan provides the $\hat{R}$ convergence diagnostic. $\hat{R}$ compares inter- and intra-chain estimates for model parameters. If these estimates do not match, then $\hat{R}$ will be > 1. Inter chain mixing can also be verified by inspecting the diagnostic plots shown in figure 16. Problems with model specification can be detected by checking HMC diagnostics. Stan has a builtin maximum treedepth to avoid infinite loops that may occur when misspecifying models. This value is set to 10 by default but complex models may also hit this limit. If that is the case, the model should be rerun with a higher maximum tree depth. Another very useful way of identifying if there are problematic regions in posterior space is the number of divergent transitions. Roughly speaking, divergent transitions indicate that regions in posterior space exist, where the local geometry caused the transition path to diverge from a path of constant energy. This can be caused by misspecifying models or a step size that is too large.

5.5 Bridges from R to Stan

The package “brms” [9] allows for nearly seamless translation of “gam”-style R model syntax to Stan code. Version 2.12.0 already contained the building blocks for performing exact and approximate Gaussian process regression for squared exponential kernels. Because “brms” uses “rstan” [25] for execution, additional Matérn kernels were not yet available. To get around this issue, the package “cmdstanr” [26] can be used as a bare bones interface to the full Stan math library. “brms” is by far the most powerful interface presented here, it is the only package that supplies builtin methods for diagnostics, visualization, and prediction of new data.
The Stan code for Gaussian Processes was structured in the following way:

```
 vector gp(vector[] x, real alpha, vector lscale, vector zgp) {
 int Dls = rows(lscale);
 int N = size(x);
 matrix[N, N] cov;
 if (Dls == 1) {
  // one dimensional or isotropic GP
  cov = cov_exp_quad(x, alpha, lscale[1]);
 } else {
  // multi-dimensional non-isotropic GP
  cov = cov_exp_quad(x[, 1], alpha, lscale[1]);
  for (d in 2:Dls) {
   cov = cov .* cov_exp_quad(x[, d], 1, lscale[d]);
  }
 }
 for (n in 1:N) {
  // deal with numerical non-positive-definiteness
  cov[n, n] += 1e-8;
 }
 return cholesky_decompose(cov) * zgp;
}
```

The function *gp* takes an array of vectors x the lengthscale *lscale* and *alpha* parameters of the kernel function and a vector *zgp* of $\mathcal{N}(0,1)$ distributed random variables. If there is only one *lscale* parameter, the covariance function $cov_{e}xp_{q}uad$ is calculated directly. If that is not the case, i.e. a multi dimensional GP needs to be calculated, use of the kernel multiplication property is made. The combined covariance can be represented as the product of the kernels for each individual dimension. After the covariance matrix is computed, a small number is added to the diagonal to guarantee positive definiteness for the next step. After this, the product of the cholesky decomposition of the covariance matrix and *zgp* is returned, because decomposition speeds up downstream computations.

For the computation of approximate GPs, the preimplemented spectral representation of the GP was used. This could only be used in combination with the RBF kernel.

6 Model Specification & Results

6.1 Variable Selection

The models were specified in two steps. First, variable selection based on significance was performed. A generalized additive model with smooth terms for every variable and a low rank Gaussian process term for the variables longitude and latitude was estimated, using additional penalization for variable selection and restricted maximum likelihood estimation. Because the total estimated degrees of freedom for temperature, distance to water, frost days, sun hours, and slope was very low, and inspection of the marginal effects plots revealed fits that were mostly linear, these variables were then no longer included as smooth terms.
Estimating a model with non smooth terms for these variables, revealed that frost-days, rain and sunhours were no longer significant at all. In total five different models were compared, including a model with a smooth interaction term for temperature and elevation. Inspection of the effect plot revealed, that there was most likely no interaction present. The final model was chosen based on AIC and BIC values for all models.

```
model <- gam(site ~ s(lon, lat, bs = "gp", m = 2) + s(dem) +
    temp + s(rain) + distance_water + sunhours + s(tpi) +
        slope, family = binomial, data = evidence,
        select = TRUE, method = "REML")
```

To then compare the different covariance functions, The AUC was calculated based on 100 cross validation runs with 80% training data and 20% test data each. The predictive performance estimates are depicted in figure 17.

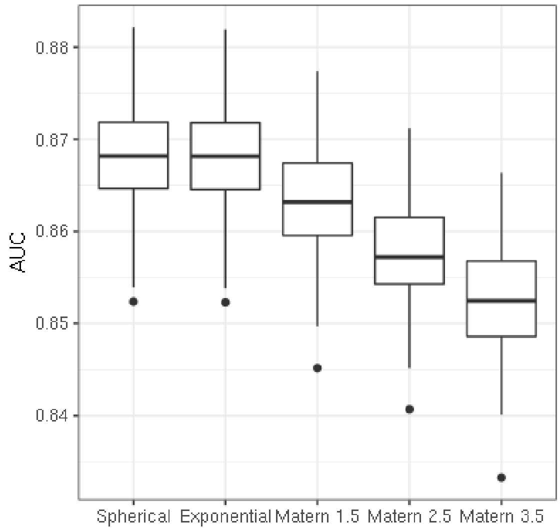


Figure 17: Performance estimates for out of sample prediction for all covariance functions. The spherical covariance was marginally better than the squared exponential function. Matérn covariances performed slightly worse, depending on the chosen κ. All models performed better than basic logistic regression with an average AUC of 0.76.

6.2 Results

The estimated smooth effects for all models were generally almost identical, the only exception being the smooth terms of the variable rain. The estimated effects of rain can be split into two groups, where the estimates under the spherical, exponential and Matérn covariance with κ of 1.5 were very similar and the estimated effects under Matérn covariance with $\kappa > 1.5$ were similar to each other, but overall different from the estimates of the other group. The overall trends stayed the same however. The estimated spatial effects were strongest for the spherical and exponential covariances. The higher κ for the Matérn covariance models, the smoother the resulting surface. An example for the effects plots is given in figure 18

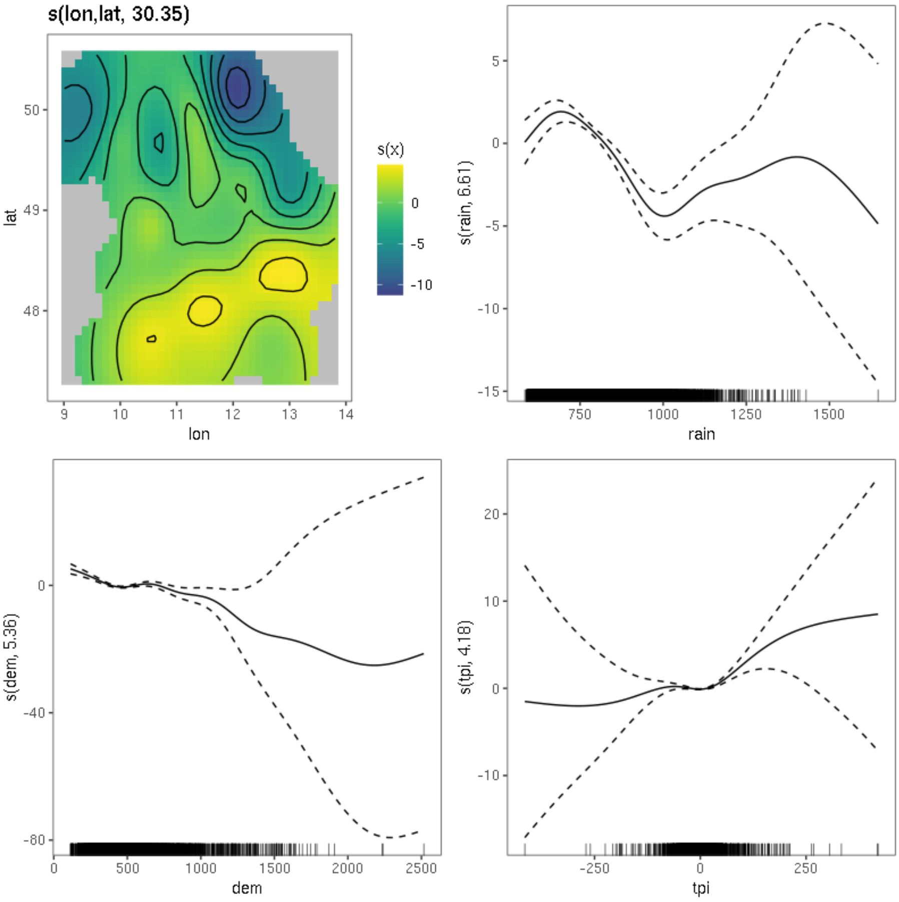


Figure 18: The estimated effects of the exponential covariance model. The estimated spatial effect was largest in the south and south east of Bavaria. Up until about 600 mm of rain per year, additional precipitation had positive effects on the predicted probability of archaeological finds, from about 800 mm onwards, additional rain had a negative effect. Increases in elevation were coupled with negative effects on the predicted outcome. Increases in TPI and, as such, increases in the height relative to the surrounding area, had a positive effect on the predicted outcome. In regions where available data points were sparse, the confidence intervals drifted apart. (The remaining plots are available in the appendix.)

The estimated coefficients for parametric terms are given in the following table. Overall, the estimated coefficients were very similar. It seems interesting that c.p. an increase in slope has a positive effect on finding a site at a given location.

	Spherical	Exponential	Matern 1.5	Matern 2.5	Matern 3.5
Intercept	-9.25	-9.19	-7.09	-0.18	-4.83
Temp	1.00	0.99	0.76	1.21	0.42
Water Dist.	-4.28e-5	-4.17e-5	-4.00e-5	-2.96e-5	-1.97e-5
Sunhours	0.28	0.24	0.14	0.16	0.19
Slope	5.60e-2	5.72e-2	6.15e-2	5.36e-2	7.87e-2

Only the estimated coefficient for temperature was not significant. Increasing the distance to the nearest water source has a negative effect on the odds of finding an archaeological site at a given location, when controlling for the other variables. An increase in sun hours had, c.p., a positive effect.

For the Bayesian estimation with Stan, models including smooth terms or the variables sun hours and TPI did not converge. Because of this, the "brms" model was specified in the following way:

```
bayesian_model <- brms::brm(site ~ gp(lon, lat, k = 30, c = 5/4) +
        distance_water + dem + temp + rain,
        family = bernoulli, data = evd_3000,
        chains = 4, cores = 4, iter = 1000,
        control = list(adapt_delta = 0.8, max_treedepth = 12))
```

These models were run with 3000 data points as evidence because of time constraints. For every model four chains were run for 500 warm up iterations and 500 sampling iterations. Additionally, two models with no covariates were run to map the estimated spatial effects. One with an isotropic kernel, the other with an anisotropic kernel. The results were not of particular interest and are depicted in the appendix.
Out of the models with Matérn kernels, only the one with a κ of 1.5 converged, and only when estimating the spatial effect alone.

6.3 Diagnostics

The Bayesian equivalent to the squared exponential covariance, generalized additive model did not converge. Three of the chains seemed to have gotten stuck, the model was either severely ill specified or otherwise problematic.

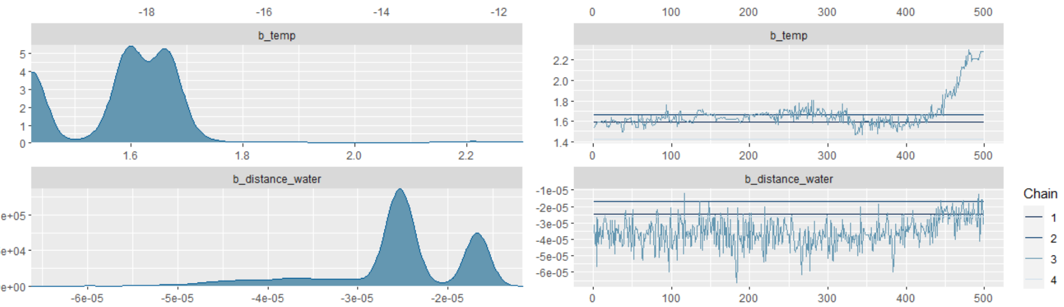
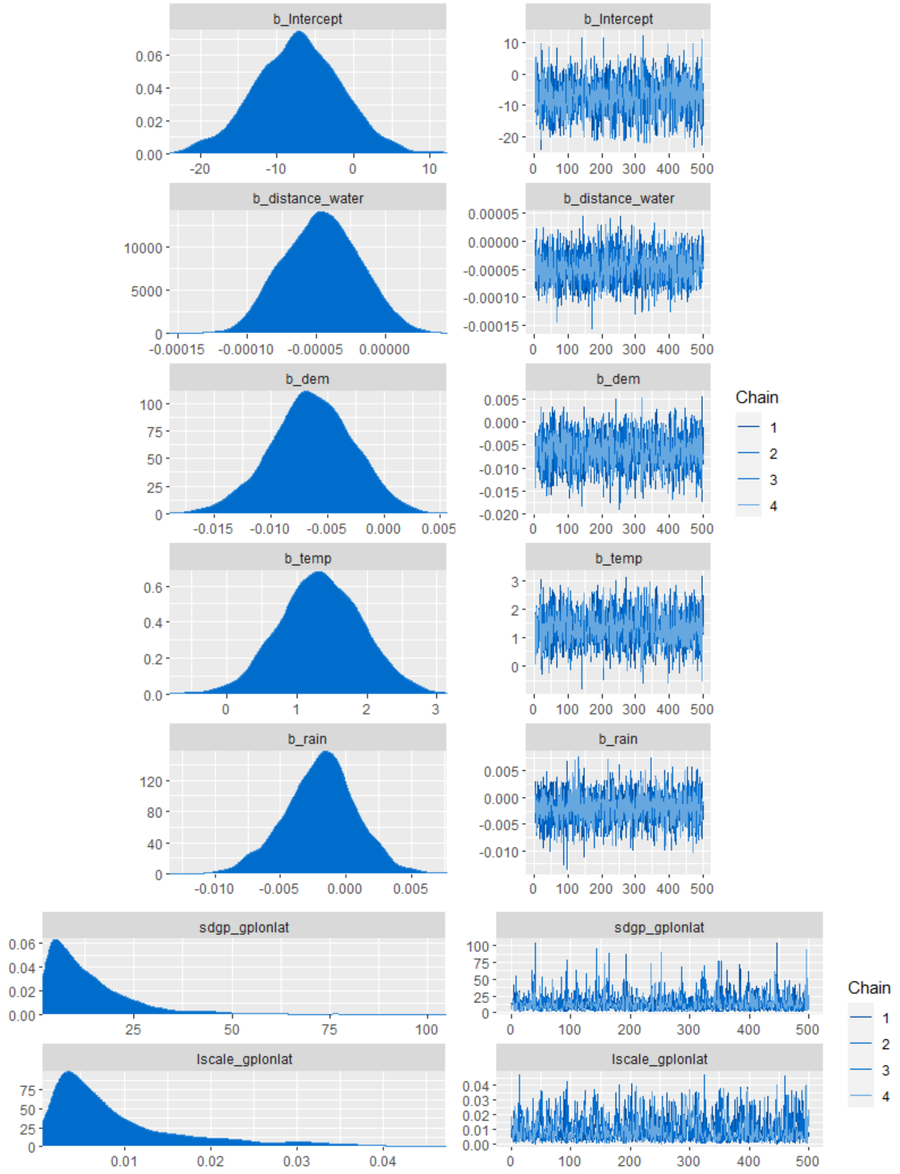


Figure 19: Two of the sample diagnostics for the Bayesian model containing all smooth terms. Three of the chains got stuck, and even the remaining chain does not exhibit convergence. As a result the posterior distribution estimates are not usable.

The model using exponential covariance and only 4 parametric terms was well behaved. The diagnostic plots are depicted below.
The top 4 plots show the posterior distributions for the parameter estimates. Here,



most likely because of the reduced sample size of 3000 points, the 95% credibility intervals for distance water, elevation and rain include the value 0. The bottom panel shows the posterior distributions for the exponential covariance function. All chains seemed to be mixing well. The models were initialized with default prior values recommended by brms.

6.4 Predictive Maps

Predictive maps were generated for all GAMs, a comparable map for the Bayesian models was only generated for the exponential covariance model. Here, 50 draws from the posterior were sampled for every one of the about 120.000 raster cells. For the Bayesian Matérn models, no maps could be generated, as the "fit and predict" method that is used to make predictions on new data for Stan models while they

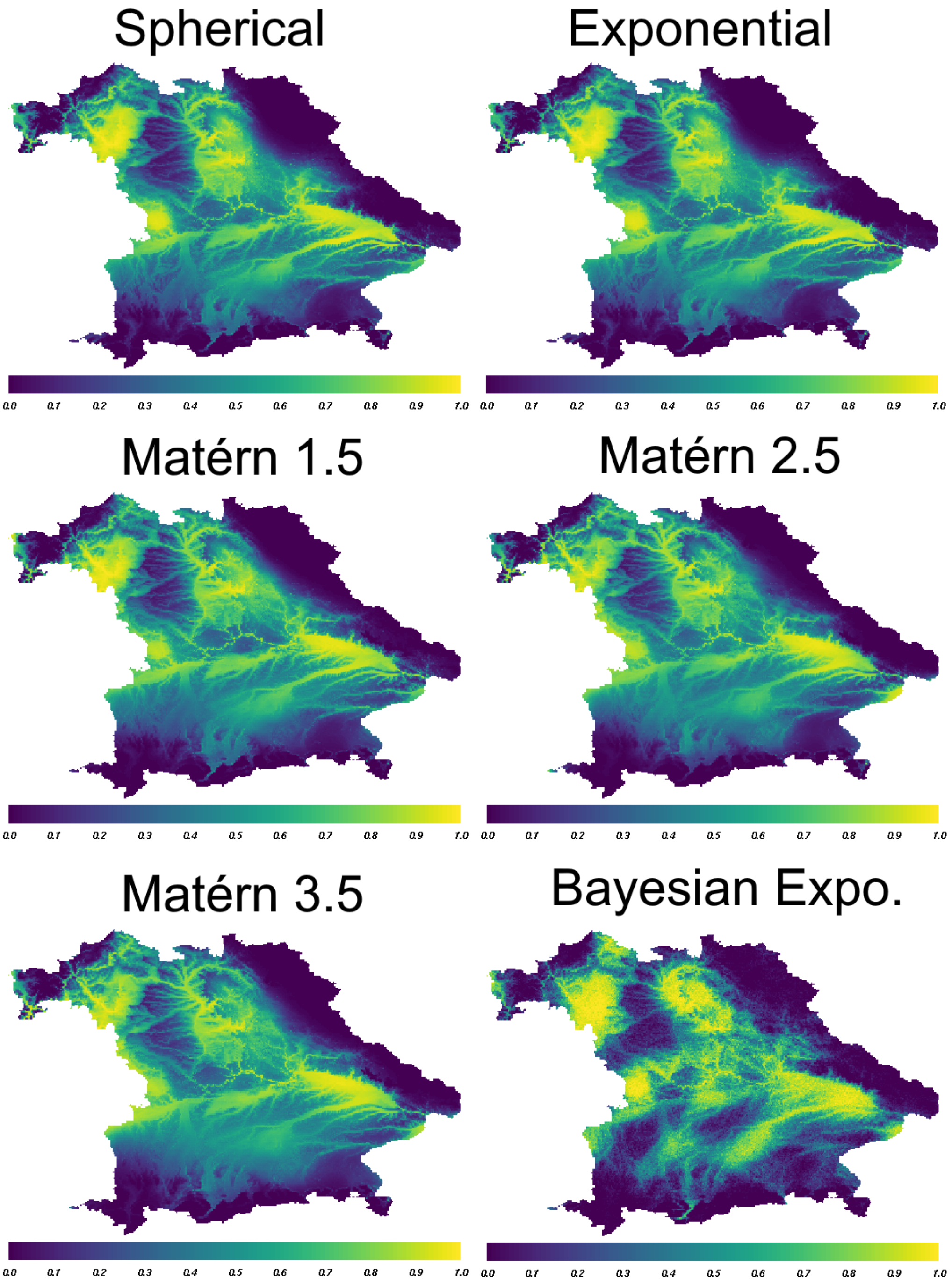


Figure 20: The generated predictive maps. The maps for spherical and exponential GAMs are nearly identical. They exhibit sharper contrast than the other models and seem to capture some of the river structure of Bavaria. The Matérn maps are getting smoother with increasing κ. The Bayesian map seems to be rougher overall, probably due to the 50 samples drawn per pixel. Increasing this number should yield smoother resulting maps overall.

are fitting did not work – most likely due to misspecification of the values to be predicted.
The Matérn κ 1.5 model did not contain any divergent transitions and the chains seemed to have converged with $\hat{R}$ values very close to 1. The means of the estimated covariance parameters were equal to $\alpha = 3.31$ and $lengthscale = 6.60^{-2}$

7 Discussion

Overall, including spatial relationships in the model increased the predictive performance of the logistic model across the board. The GAMs estimated with mgcv took about one minute of run time per model. Exact Bayesian GP models using the full data set would have taken multiple weeks of run time, but approximating them with the spectral density representation decreased run time drastically, making it feasible to use the full data set for inference.
Future interesting work may be focused on applying the spectral density approximation for Matérn covariance models, as this seemed to be the main limiting factor to diagnosis and specification of these models. Splines could also be used for Bayesian modeling, as here the implementation in "brms" is already sufficient but the issues with the model that included splines could not be treated in time.
It should be mentioned that the flexibility of full Bayesian inference is a deciding factor. This is further amplified by not having to spend large amounts of time on tuning the sampler when using Stan. For most models the available interfaces to Stan are sufficiently powerful, but if special functions (such as Matérn kernels) are needed directly, using Stan through the command line or terminal is the best way of avoiding compatibility issues. Another way of directly decreasing computational cost could be to group the data up in a raster grid of lower resolution and then use Poisson regression combined with Gaussian Processes.
In the end, this thesis only scratches the surface of what is possible, as expanding the analysis to include non-stationarity would be another way of continuing research.

References

[1] Simo Särkkä Arno Solin. "Hilbert Space Methods for Reduced-RankGaussian Process Regression". In: (2019). URL: arXiv:1401.5508.

[2] Michael Betancourt. "A Conceptual Introduction to Hamiltonian Monte Carlo". In: (2018). P. 3. URL: arXiv:1701.02434v2.

[3] Michael Betancourt. "A Conceptual Introduction to Hamiltonian Monte Carlo". In: (2018). P. 19. URL: arXiv:1701.02434v2.

[4] Michael Betancourt. "A Conceptual Introduction to Hamiltonian Monte Carlo". In: (2018). P. 23. URL: arXiv:1701.02434v2.

[5] Michael Betancourt. "A Conceptual Introduction to Hamiltonian Monte Carlo". In: (2018). P. 25. URL: arXiv:1701.02434v2.

[6] Michael Betancourt. *Robust Gaussian Processes in Stan.* Accessed: 2020-04-22. URL: https://betanalpha.github.io/assets/case_studies/gp_part1/part1.html.

[7] Michael Betancourt. *Robust Gaussian Processes in Stan.* Section 1.2; Accessed: 2020-04-22. URL: https://betanalpha.github.io/assets/case_studies/gp_part1/part1.html.

[8] A. Brenning. "Spatial prediction models for landslide hazards: review, comparison and evaluation". In: (2005).

[9] Paul-Christian Bürkner. *brms: Bayesian Regression Models using 'Stan'*. R package version 2.12.0. 2019. URL: https://cran.r-project.org/web/packages/brms.

[10] C. Williams C. Rasmussen. "Gaussian Processes for Machine Learning". In: (2006). P. 14. URL: http://www.gaussianprocess.org/gpml/chapters/RW.pdf.

[11] C. Williams C. Rasmussen. "Gaussian Processes for Machine Learning". In: (2006). P. 57. URL: http://www.gaussianprocess.org/gpml/chapters/RW.pdf.

[12] Haizhang Zhang Charles Micchelli Yuesheng Xu. "Gaussian Processes for Machine Learning". In: (2006). Section 3. URL: http://jmlr.csail.mit.edu/papers/volume7/micchelli06a/micchelli06a.pdf.

[13] David Barber Christopher Williams. "Bayesian ClassificationWith Gaussian Processes". In: (1998). Vol 20. No. 12, Section 2. URL: https://publications.aston.ac.uk/id/eprint/4491/1/IEEE_transactions_on_pattern_analysis_20%5C%2812%5C%29.pdf.

[14] *Daten des Deutschen Wetterdiensts: Frosttage.* Frostdays annual map normals 1971 30. URL: https://maps.dwd.de/geoserver/web/wicket/bookmarkable/org.geoserver.web.demo.MapPreviewPage?0 (visited on 04/22/2020).

[15] *Daten des Deutschen Wetterdiensts: Sonnenstunden.* dwd:SDMS_17_1961_30. URL: https://maps.dwd.de/geoserver/web/wicket/bookmarkable/org.geoserver.web.demo.MapPreviewPage?0 (visited on 04/22/2020).

[16] David Duvenaud. *The Kernel Cookbook: Advice on Covariance functions.* 2020-04-22. URL: https://www.cs.toronto.edu/~duvenaud/cookbook/.

[17] *Eisenzeit Datenbank.* URL: https://pma.gwi.uni-muenchen.de:8888/sql.php?server=9&db=vfpa_eisenzeit&table=fender_2017 (visited on 04/20/2020).

[18] Ludwig Fahrmeir et al. *Statistik: Der weg zur Datenanalyse.* Springer-Verlag, 2016.

[19] Peer Fender. “Bayern in der Vorgeschichte-Eine GIS-gestützte Analyse der Siedlungslandschaft und der Einsatz von Open Data in der Archäologie”. In: (2017).

[20] Fick, S.E., R.J. Hijmans. *Worldclim 2: New 1-km spatial resolution climate surfaces for global land areas*. 2017. URL: `http://worldclim.org/version2` (visited on 04/22/2012).

[21] *Gewässershapefile Deutschland*. URL: `https://biogeo.ucdavis.edu/data/diva/wat/DEU_wat.zip` (visited on 04/22/2020).

[22] Torsten Hothorn Göran Kauermann. *Skript aus der Vorlesung Statistik IV*. Sommersemester 2019.

[23] Berthold Horn. “Hill Shading and the Reflectance Map”. In: (1981).

[24] Berthold Horn. “Hill Shading and the Reflectance Map”. In: (1981). Paraphrased from description of Fig. 4.

[25] et al. Jiqiang Guo. *lightweight interface to Stan for R users*. R package version 2.19.3. 2019. URL: `https://cran.r-project.org/web/packages/rstan`.

[26] Rok Cesnovar Jonah Gabry. *rstan: R Interface to Stan*. R package version 0.0.0.9000. 2020. URL: `https://mc-stan.org/cmdstanr/articles/cmdstanr.html`.

[27] Christine Jula. “Räumliche Modelle”. In: (2015). Abschnitt 4. URL: `https://epub.ub.uni-muenchen.de/25585/1/MA_Jula.pdf`.

[28] Daniel G. Krige. “A statistical approach to some basic mine valuation problems on the Witwatersrand.” In: (1951). 52 (6) S. 119–139.

[29] Ben Lambert. *A Student’s Guide to Bayesian Statistics*. P. 292. Sage, 2018.

[30] Guy Lebanon. *The Exponential Family of Distributions and Logistic Regression*. Accessed: 2020-04-22. URL: `http://theanalysisofdata.com/notes/expFamily.pdf`.

[31] *Lines and Circles and Logistic Regression*. URL: `http://blog.data-miners.com/2014/03/lines-and-circles-and-logistic.html` (visited on 09/10/2019).

[32] Stefan Lang Ludwig Fahrmeir Thomas Kneib. *Regression: Modelle, Methoden & Anwendungen*. Second Edition, P. 189. Springer-Verlag, 2009.

[33] Stefan Lang Ludwig Fahrmeir Thomas Kneib. *Regression: Modelle, Methoden & Anwendungen*. Second Edition, P. 292, Chapter 7.1. Springer-Verlag, 2009.

[34] Stefan Lang Ludwig Fahrmeir Thomas Kneib. *Regression: Modelle, Methoden & Anwendungen*. Second Edition, P. 317, Chapter 7.1. Springer-Verlag, 2009.

[35] Stefan Lang Ludwig Fahrmeir Thomas Kneib. *Regression: Modelle, Methoden & Anwendungen*. Second Edition, P. 296, Chapter 7.1. Springer-Verlag, 2009.

[36] Andrew Gelman Matthew D. Hoffman. “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo”. In: (2011). URL: `https://arxiv.org/abs/1111.4246`.

[37] Andrew Gelman Matthew D. Hoffman. “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo”. In: (2011). P. 25. URL: `https://arxiv.org/abs/1111.4246`.

[38] Andrew Gelman Matthew D. Hoffman. “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo”. In: (2011). P. 5. URL: `https://arxiv.org/abs/1111.4246`.

[39] C. Cressie Noel A. *Statistics for Spatial Data*. Wiley Series in Probability and Statistics. Wiley, 1993.

[40] Grace Wahba Peter Craven. “Smoothing Noisy Data with Spline Functions”. In: (1979). 31,377-403 (1979).

[41] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria, 2014. URL: `http://www.R-project.org/`.

[42] Jacob van Etten et al. Robert J. Hijmans. *raster: Geographic Data Analysis and Modeling.* R package version 3.0-2. 2019. URL: `https://cran.r-project.org/web/packages/raster/`.

[43] Jannes Muenchow Robin Lovelace Jakub Nowosad. *Geocomputation with R.* 2019. URL: `https://geocompr.robinlovelace.net/` (visited on 09/10/2019).

[44] S. Sekar S. Karunanithi S. Chakravarthy. "A Study on Second-Order Linear Singular Systems using Leapfrog Method". In: (2014). Vol. 5, Issue 5. URL: `https://www.ijser.org/researchpaper/A-Study-on-Second-Order-Linear-Singular-Systems-using-Leapfrog-Method.pdf`.

[45] Edward Greenberg Siddhartha Chib. "Understanding the Metropolis-Hastings Algorithm". In: (1995). Vol. 49, No. 4 pp. 327-335. URL: `arXiv:1701.02434v2`.

[46] Stan Development Team. *Stan Modeling Language Users Guide and Reference Manual.* 2018. URL: `http://mc-stan.org`.

[47] Stan Development Team. *Stan Reference Manual.* 2018. URL: `https://mc-stan.org/docs/2_18/reference-manual/effective-sample-size-section.html`.

[48] Michael Stein. *Interpolation of Spatial Data.* Springer Series in Statistics. P. 30. Springer, 1999.

[49] Kenneth Tay. *Bayesian Interpretation of Ridge Regression.* Accessed: 2020-04-22. URL: `https://statisticaloddsandends.wordpress.com/2018/12/29/bayesian-interpretation-of-ridge-regression/`.

[50] Waldo. Tobler. "A computer movie simulating urban growth in the Detroit region". In: (1970). 46, P. 234-240. URL: `https://www.jstor.org/stable/143141`.

[51] Robert Tibshirani Trevor Hastie. "Generalized Additive Models". In: (1986). URL: `http://www.uvm.edu/pdodds/files/papers/others/1986/hastie1986a.pdf`.

[52] Simon Wood. *Generalized Additive Model Selection.* R package version 1.8-31. 2019. URL: `https://astrostatistics.psu.edu/su07/R/library/mgcv/html/gam.selection.html`.

[53] Simon Wood. *Generalized Additive Models: An Introduction with R.* Texts in Statistical Science. Chapman & Hall, 2006.

[54] Simon Wood. *Generalized Additive Models: An Introduction with R.* Texts in Statistical Science. P. 168. Chapman & Hall, 2006.

[55] Simon Wood. *Generalized Additive Models: An Introduction with R.* Texts in Statistical Science. P. 175. Chapman & Hall, 2006.

[56] Simon Wood. *Generalized Additive Models: An Introduction with R.* Texts in Statistical Science. P. 180. Chapman & Hall, 2006.

[57] Simon Wood. *mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation.* R package version 1.8-31. 2019. URL: `https://cran.r-project.org/web/packages/mgcv`.

8 Appendix

The effect plots for the non-exponential covariance GAMs:

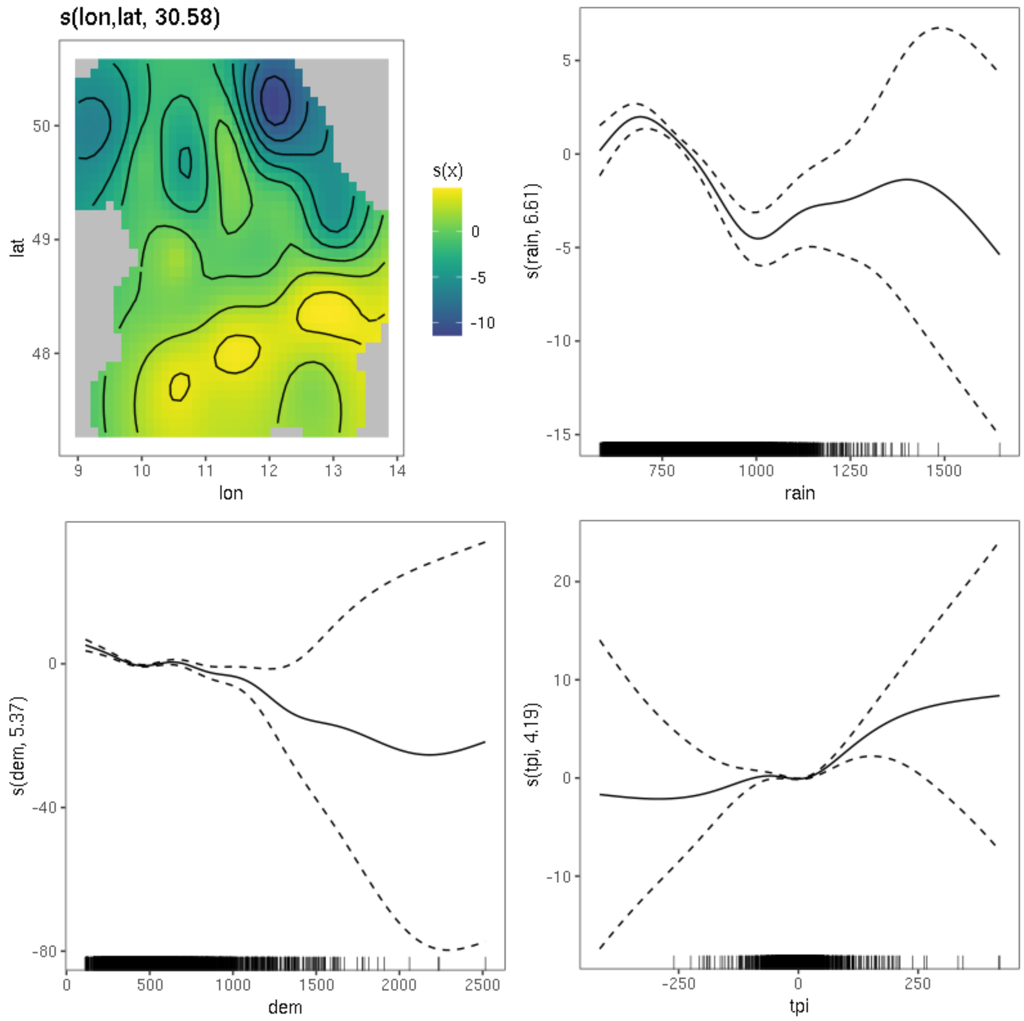


Figure 21: Effects of estimated in the spherical covariance model.

Contents of the CD: The contents of the CD are mirrored on Github and are ordered in the following manner:
Daten: Contains the raster data and data provided by Fender
Misc: Contains all R scripts and files that were generated while working on the thesis but were not of importance for the final result.
Models: Contains all code relevant for the modeling of the data
Plots: Contains all visualizations and code needed to generate them
Preprocessing: Contains scripts for data preprocessing
Results: Contains RDS files with model results. LaTeX: Contains Zip file with Tex files to generate this document.

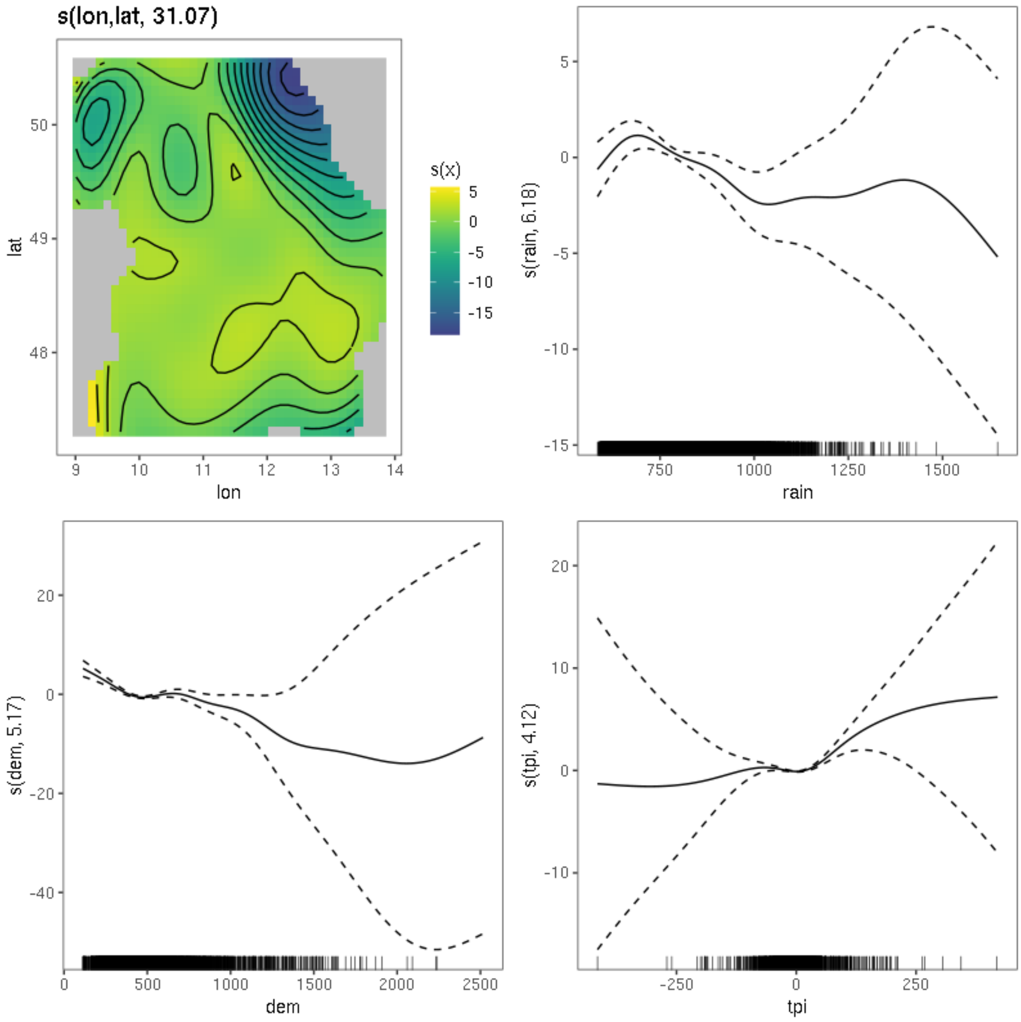


Figure 22: Effects of estimated in the Matérn κ 1.5 covariance model.

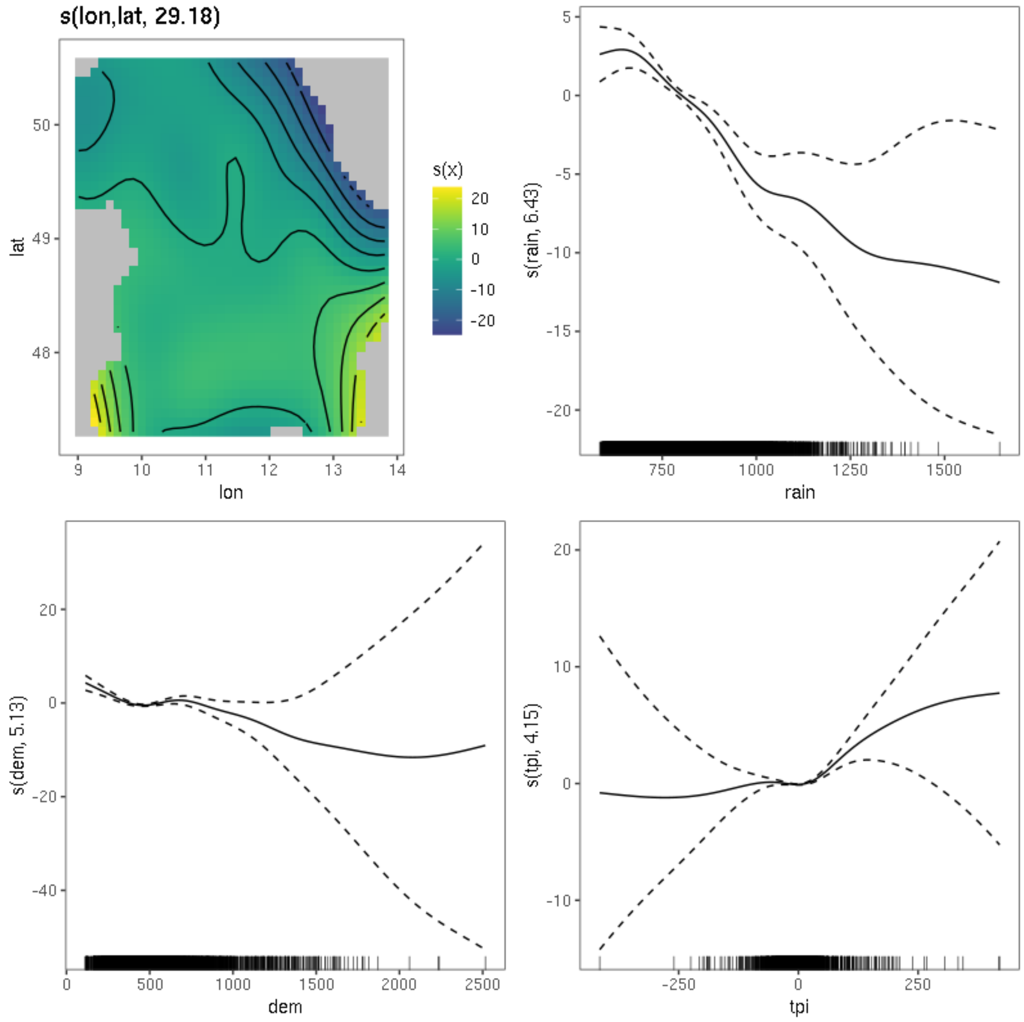


Figure 23: Effects of estimated in the Matérn κ 2.5 covariance model.

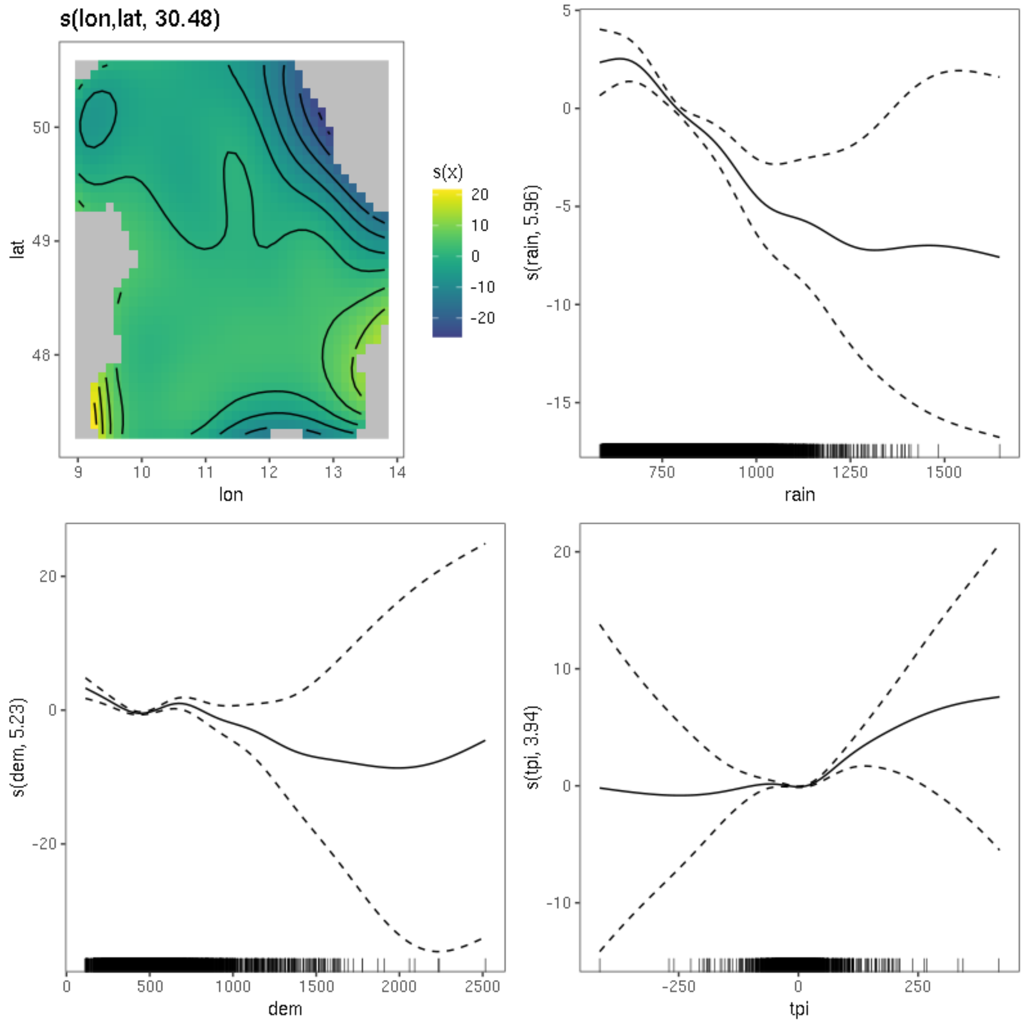


Figure 24: Effects of estimated in the Matérn κ 3.5 covariance model.

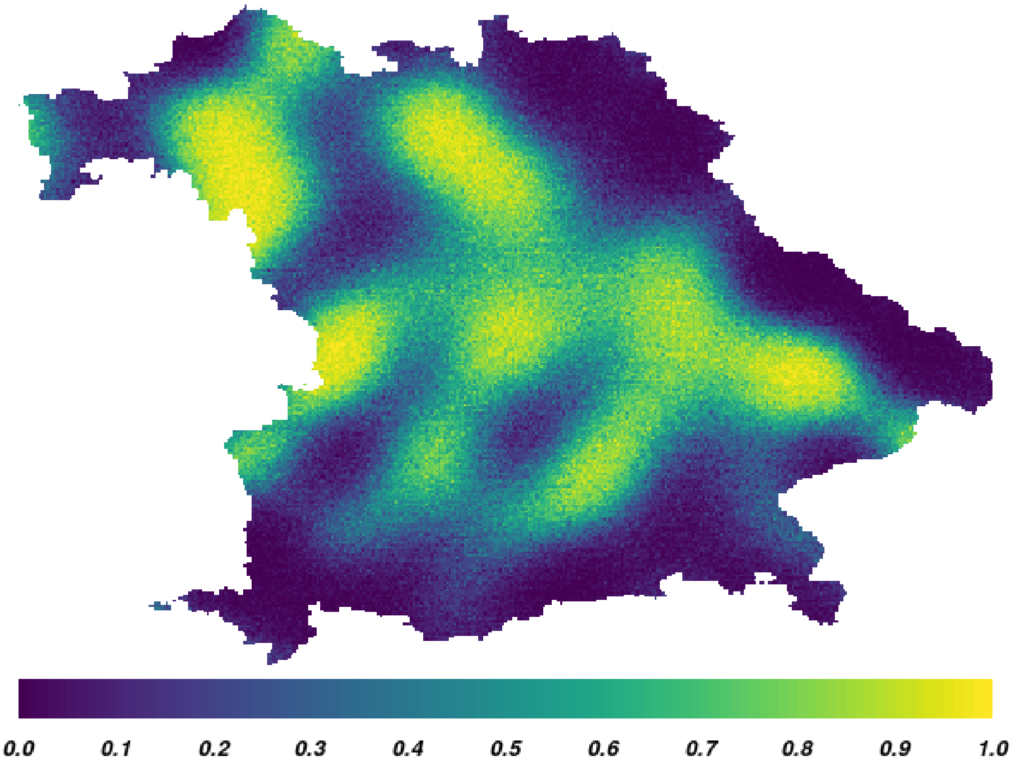


Figure 25: Spatial effect of the isotropic model mentioned in section 6

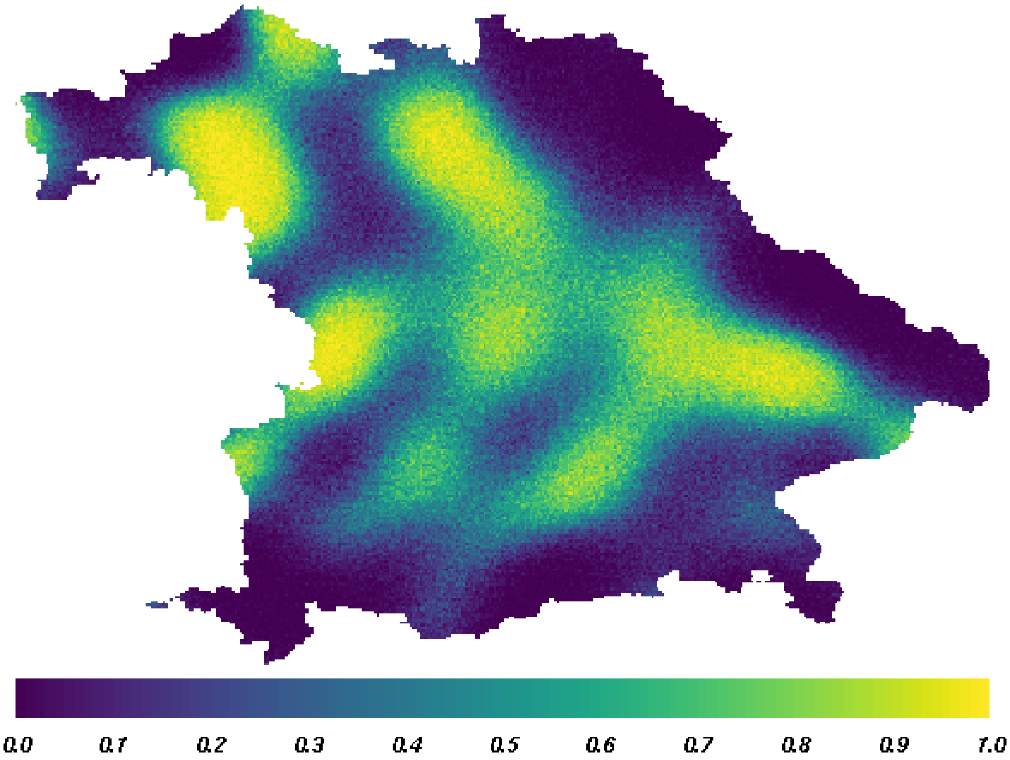


Figure 26: Spatial effect of the anisotropic model mentioned in section 6

Eigenständigkeitserklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit eigenständig und ohne fremde Hilfe verfasst, keine anderen als die angegebenen Quellen verwendet und die den benutzten Quellen entnommenen Passagen als solche kenntlich gemacht habe. Diese Arbeit ist in dieser oder einer ähnlichen Form in keinem anderen Kurs und/oder Studiengang als Studien- oder Prüfungsleistung vorgelegt worden.
Hiermit stimme ich zu, dass die vorliegende Arbeit von der Prüferin/ dem Prüfer in elektronischer Form mit entsprechender Software überprüft wird.

München, den April 29, 2020 . . . .

Statement of authorship

I hereby declare that the thesis I am submitting is entirely my own original work except where otherwise indicated.
I am aware of the University's regulations concerning plagiarism, including those regulations concerning disciplinary actions that may result from plagiarism. Any use of the works of any other author, in any form, is properly acknowledged at their point of use. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after esamination, to enable comparison with future theses submitted.

Munich, April 29, 2020 . . . .