Emulators for dynamic vegetation models - Supervised learning in large data sets

Olof Olsson

Master's thesis
2017:E66

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

Abstract

The observed and expected changes in the environment due to human actions implies risks that future food production will be insufficient. Predicting the impact these changes have on the agricultural system could be beneficial by allowing for proactive mitigating efforts. The prediction of these impacts often involve large computer programs that simulate the behavior of the environment. By implementing a statistical representation of the simulator, called an emulator, our hope is that these predictions could be obtain at a lower computational cost. This master thesis has implemented and evaluated a Gaussian process emulator for a vegetation model that is used for predicting the annual production of spring wheat based on climate data at different locations around the world. The problem of accurately modeling the simulator using a Gaussian process approach was split into two parts. The first part was to model the average yield at each location given average climate input at that location. The second part was to model the yield at a specific year for a location given the average yield at that location and the climate input anomalies during that year. The results was far from satisfactory and a more complex approach is probably needed before the emulator can be of any practical use. Based on our findings, possible extensions that might improve results are discussed.

Keywords: Gaussian process emulator, DGVM, emulation.

Acknowledgments

This thesis was conducted at the division of mathematical statistics at Lund university in collaboration with the department of physical geography and ecosystem science at Lund university. I am thankful for all the received support and guidance, both from my supervisors and from other employees at the university.

A special thanks should be directed to Johan Lindström who supervised the work and for his patience, support and exceptional guidance.

Lastly, I want to thank my family and friends for their support, both during the writing of this master thesis and during my entire period of studies.

Lund November 21, 2017

Contents

1 Introduction

1.1 Problem description

General concepts are provided in sections 1.2 - 1.3. The problem is described in section 1.4. Section 1.5 - 1.6 gives an introduction to emulators and discusses different emulator strategies. The main purpose of the thesis is given in section 1.7.

1.2 Simulator

A simulator is essentially a complex model that is constructed to imitate a real-world process over time. These models are usually large computer programs that take hours to run (O'Hagan, 2006). The simulator can be deterministic, i.e. the simulator gives the same output every time for the same input or it can give stochastic outputs generated from a random number seed.

1.3 Dynamic global vegetation models (DGVM)

A DGVM is in essence a simulator used to model different natural ecosystems and their response to climate change (Hillel, 2015, page 180). The word dynamic means that the computer program operates iteratively to model complex systems that are evolving over time (O'Hagan et al., 2009). By spatially dividing the earth's surface into a grid the DGVM may be used to simulate the behaviors of vegetation at a particular area or region. An error between the output from the computer model and the real-world phenomena will inevitably exist (O'Hagan, 2006).

1.4 Problem description

The increase in world population along with observed and expected changes in the environment due to human actions implies risks that future food production will be insufficient. Many initiatives have started in order to construct large and informative DGVMs for agricultural systems that can predict how food production will be affected by future climate. One of these initiatives is AgMIP which is a large multinational collaboration to improve agricultural predictions when the environment is affected by a diversity of climate changes (AgMIP, 2017). A problem AgMIP faces is that their agricultural models, which provides simulated harvests, are computationally demanding. Thus creating predictions where the impact of many potential future climate scenarios are considered will be expensive.

1.5 Emulator

One option to reduce the computations needed to retrieve the predictions from the DGVM is to construct an emulator. An emulator is a common name for a statistical representation of a simulator that is constructed using a training sample of simulator runs (O'Hagan et al., 2009). The simulator is viewed as a unknown mathematical function $f(\cdot)$ of its input although the outputs can be obtained by running the computer code. The idea of the emulator is that once it is constructed it can be used to cheaply obtain the output of the simulator without actually running it, drastically reducing the execution time for obtaining these results (O'Hagan, 2006). An error exists between what the emulator returns for a given input and what the simulator would have returned for the exact same input. A good emulator gives accurate predictions of the simulator as well as a prediction of the uncertainty related to using it. The concept of an emulator is illustrated in figure 1.

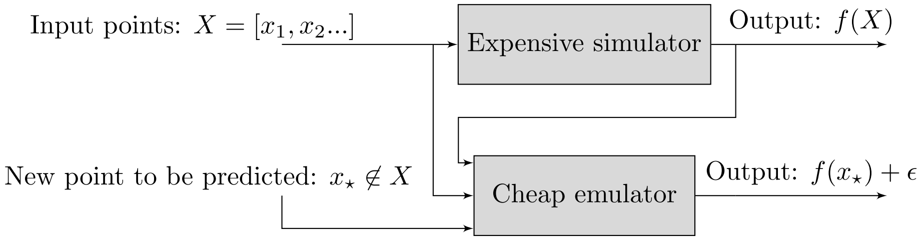
Input points: $X = [x_1, x_2...]$
Expensive simulator
Output: $f(X)$
New point to be predicted: $x_\star \notin X$
Cheap emulator
Output: $f(x_\star) + \epsilon$

Figure 1: The idea behind an emulator is to use a training sample of simulator runs to construct the emulator. The emulator can later be used to cheaply obtain future simulator outputs. The difference, or error, between the emulator and simulator output for a point $x_\star$ is denoted ϵ.

1.6 Emulator strategies

The problem of finding the optimal emulator to represent the behavior of a given simulator is difficult. This relates to the famous "no free lunch theorem" by Wolpert (1996) which implies that there is no universally good model that works for all problems (Murphy, 2012, page 24). As a consequence of this theorem there exist a variety of different models to choose from. Common emulator strategies include pattern scaling (Castruccio et al., 2014), empirical orthogonal function (Castruccio et al., 2014), regression models (O'Hagan, 2006), neural networks (O'Hagan, 2006) and Gaussian processes (O'Hagan, 2006). Finding the best strategy among all these possibilities is called the model selection problem. The solution usually involves trying each strategy and compare their results respectively (Murphy, 2012, page 156). In practice this procedure is so time consuming that it becomes practically impossible to perform. Instead experience and knowledge about the problem are used to determine a suitable trade-off between which models to try and and their likelihood of success.

When searching for the best emulator strategy the mathematical structure of the actual simulator can be investigated and analyzed. The benefits of this analysis might be that relationships can be discovered and incorporated in the model, resulting in a more favorable emulator (Kennedy and Ohagan, 2001), sometimes called "grey box" models. However, the implementation of "grey box" models are often much more complex (Kennedy and Ohagan, 2001). An alternative is to treat the model as a "black box" where only the input and outputs are analyzed.

1.7 Purpose of thesis

The purpose of this thesis is to implement and evaluate a Gaussian process emulator for the yearly production of spring wheat modeled by one DGVM at different locations and using different climate data as input. The accuracy, benefits and drawbacks of this method will be examined and discussed. The thesis will also investigate the practical usability of the implemented emulator and give recommendations for further research.

2 Vegetation model - LPJ-GUESS

LPJ-GUESS or the Lund-Potsdam-Jena General Ecosystem Simulator is a DGVM originally developed by Ben Smith at Lund University (Smith et al., 2001). LPJ-GUESS models the structure and dynamics of terrestrial ecosystems at different areas and it can be used to predict a variety of ecological processes including annual harvest yield[1]. LPJ-GUESS spatially divides Earth's surface into a grid and then models the vegetation within each grid cell based on climate, CO_2 levels and other inputs. The grid specific inputs needed to predict the annual production of spring wheat are described below[2].

2.1 Time series input for each grid cell

Following subsections provides a brief description of the inputs needed at each grid cell for LPJ-GUESS to predict wheat yield at that grid cell.

2.1.1 Surface downwelling shortwave radiation flux (rsds)

This will be denoted "radiation" or "R" in the plots, measured in W/m^2. The input describes the daily radiative energy that reaches Earth's surface per time and surface unit at a specific location. The shortwave (280 to 2800 nm) radiation flux is often the most important quantity when calculating the total available energy at the surface. (Geiger et al., 2008). The shortwave radiation greatly impacts certain vegetation processes such as transpiration, evaporation and photosynthesis which profoundly affects the outcome of an agricultural systems (Klassen and Bugbee, 2005). Shortwave radiation is highly variable and it is especially sensitive to solar zenith angle, i.e. season, and clouds stopping the radiation from reaching the surface (Klassen and Bugbee, 2005).

2.1.2 Precipitation flux

Denoted "precipitation" or "P" in the plots, measured in $kg/(m^2)$. Precipitation describes all the aqueous (both liquid and solid) particles that fall from the atmosphere to the surface of the earth. Precipitation is generally beneficial for plants. Excessive rainfall can however drown plant roots and cause significant soil erosion. The agricultural effect of precipitation also depends on the current temperature. For example, rainfall or drizzle during very cold temperatures can damage plants (NC-University, 2013).

[1] More information about LPJ-GUESS can be found at http://iis4.nateko.lu.se/lpj-guess/index.html and http://iis4.nateko.lu.se/lpj-guess/guess.pdf

[2] LPJ-GUESS also needs global non-climate parameters such as global CO_2 concentration and fertilization supply. For a full list see http://www.agmip.org/wp-content/uploads/2016/01/GGCMI_phase_2_CTWN_protocol_v75.pdf

2.1.3 Surface air temperature at 2 meter (tas)

Denoted ”temperature” or ”T” in the plots, measured in Kelvin. The air temperature influences photosynthesis and affects soil temperature. The optimal air temperature is different for different plants. Too high or too low temperatures can affect growth indirectly by causing drought or frost damages (NC-University, 2010).

2.2 Output - Spring wheat

LPJ-GUESS can be used to predict a variety of different agricultural processes such as crop yield, CO_2 uptake, optimal harvest date and start of growing season for different crops. Here the annual yield of spring wheat is used as a test case for the emulators. Spring wheat is wheat that is sown in the spring and harvested in late summer or early fall (Merriam-Webster, 2017) The annual yield of spring wheat from LPJ-GUESS is measured in $ton/(ha \cdot year)$, dry matter.

3 Data

The available data contained inputs and corresponding outputs of spring wheat from 196 locations scattered around the world, predicted using LPJ-GUESS. See figure 2.

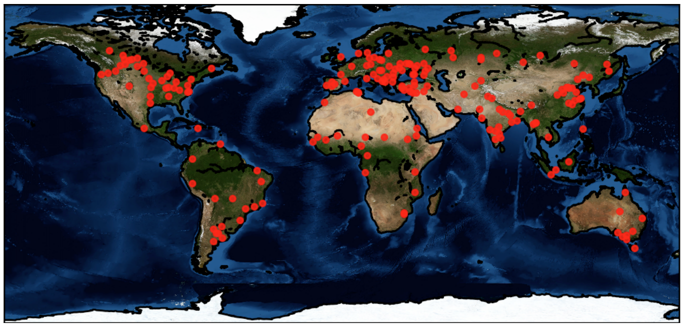

Figure 2: The 196 different locations where simulator input and output were available.

The inputs for LPJ-GUESS at each of these 196 different locations mainly consisted of three 31 year long time-series containing daily measurements of radiation, temperature and precipitation[3]. LPJ-GUESS was executed with the inputs needed to predict the annual yield of a spring wheat variety called T0P0C480/TeSW60. This output is collected when LPJ-GUESS has zero temperature difference from normal (T0), the change in precipitation from normal is zero (P0), $60kgN/ha$ fertilization is applied and the concentration of CO_2 in the atmosphere is 480 ppm (C480). Thus representing present day climate conditions but an elevated CO_2 concentration. The ultimate aim would be to fit the emulator using a variety of different future climate scenarios. For simplicity this particular output will only be called "spring wheat" or "yield" and is considered the response variable in this thesis. Table 1 displays a summary of the collected inputs and outputs.

Variable	Type	Min	Mean	Max	Unit	Plotted in figure
Spring wheat	output	0.0	1.57	10.5	Annual ton/ha	24
Precipitation	input	0	0.000025	0.0034	kg/m^2	25
Radiation	input	0	178	411	W/m^2	26
Temperature	input	228	288	316	Kelvin	27

Table 1: Describing input and output variables gathered from LPJ-GUESS. The plots displays these time-series for one grid cell.

[3] The inputs were collected from the AgMERRA data set from 1980-2010, available at: https://data.giss.nasa.gov/impacts/agmipcf/agmerra/

4 Theory - Gaussian process emulator

4.1 Definition

A GP-emulator used to model an unknown function $f(\cdot)$ typically consists of two to three components. A mean value that captures large scale behaviors, a Gaussian process[4] that represents the variability of the unknown function and white noise which describes the differences between the GP and the computer model it tries to emulate. The GP emulator can thus be described by:

$$y(x_i) = \underbrace{\mu(x_i) + \eta(x_i)}_{f(x_i)} + \epsilon_i \tag{1}$$

Where $\eta \sim N(0, K)$ and $\epsilon \sim N(0, \sigma_n^2 I)$. The observations $y(x_i)$ are Gaussian with mean $\mu(x_i)$ and a covariance matrix where the noise variance (σ_n^2) has been added to the diagonal of K (Roberts et al., 2012):

$$y(x) \sim N(\mu(x), K + \sigma_n^2 I), \tag{2}$$

where I is an identity matrix. To effectively apply Gaussian processes for regression one important assumption for the underlying function $f(\cdot)$ is required: It must have similarity between data-points. This basically means that if two x-values (x_1, x_2) are close in x-space then their corresponding values $(f(x_1),f(x_2))$ should be close (Rasmussen and Williams, 2008, page 79). The main part of applying a Gaussian process emulator is choosing the mean and covariance functions used to represent prior knowledge of the unknown function. Each mean and covariance function may also depend on parameters which needs to be estimated (Rasmussen and Williams, 2008, page 106). The model selection problem for Gaussian processes thus includes both finding the optimal values for these parameters for a specific choice of mean and covariance function and to compare different choices of mean and covariance functions (Rasmussen and Williams, 2008, page 106).

4.2 Mean function

The mean function should represent prior expectations of the underlying large scale mean structure of the data and the choice should be driven by both experience and simplicity (O'Hagan et al., 2009). It is not uncommon to choose mean function $\mu(x) = 0$ and let the covariance function try to model the mean behavior as well (Murphy, 2012, page 516). However this approach has two main disadvantages: 1) if the process is used to predict points in regions far from the training data, the lack of the mean function will lead to worse forecasts (Roberts et al., 2012). 2) The covariance function is forced to capture

[4] General properties of Gaussian processes can be found in Lindgren et al. (2014, page 111-112)

large scale structures, reducing its ability to model small scale behaviors in f (Roberts et al., 2012). Common approaches include no mean, constant mean (Roberts et al., 2012) and linear mean, i.e $\mu(x) = x\beta$ (O'Hagan et al., 2009).

4.3 Covariance function

4.3.1 Definition and requirements for a valid covariance function

A covariance function for any stochastic process is defined as (Lindgren et al., 2014, page 21):

$$k(t, t') = C[X(t), X(t')] \tag{3}$$

A common name for a function k of two arguments mapping a pair of inputs $t \in T$, $t' \in T$ into $\mathbb{R}$ is a kernel (Rasmussen and Williams, 2008, page 80). An arbitrary kernel $k(t, t')$ is in general not a valid covariance function. The kernel must fulfill some requirements before it can serve as a covariance function. First of all a covariance function must be a symmetric function in $\mathbb{R}$, $(k(t, t') = k(t', t))$ (Rasmussen and Williams, 2008, page 80). The second requirement is that the covariance matrix corresponding to a covariance function needs to be positive semidefinite (Rasmussen and Williams, 2008, page 80). This requirement implies that the kernel needs to be a positive semidefinite function (Rasmussen and Williams, 2008, page 80):

$$\int k(t, t')f(t)f(t')dtdt' \geq 0 \quad \text{where} \int f^2(t)dt < \infty \quad \text{and } f(t) \neq 0 \tag{4}$$

Since these requirements are quite tolerant many different covariance functions exists. Different covariance functions can be combined since the sum and/or product of two valid covariance kernels is a valid covariance kernel. (Rasmussen and Williams, 2008, page 95). The covariance function can also be designed to have certain properties which are explained below.

4.3.2 Special cases

To simplify understanding the covariance function is rewritten as $k(t't') = k'(r(t, t'))$.

Stationary

A Stationary covariance function only depends on the difference between t and t', i.e. $r(t, t') = t - t'$, these covariance functions are invariant to translations. Valid stationary covariance functions can be created with the help of Bochner's theorem, see Wilson (2013, section 3) for a detailed explanation.

Isotropic

A stationary covariance function that only depends on the distance in input space, i.e $r(t, t') = |t - t'|$ is called isotropic. Isotropic covariance functions are invariant to all rigid motions, i.e to both translations and rotations (Rasmussen and Williams, 2008, page 80).

Anisotropic

A stationary anisotropic covariance functions behaves differently for different directions of the input space. They can be created from an isotropic covariance function by modifying $r(t, t')$. For example; using a linear transformation i.e $r(t, t') = |A(t - t')|$ (Wackernagel, 2010, chapter 9) or by setting $r^2(t, t') = (t-t')^T M(t-t')$ for some positive semidefinite matrix $M = A^T A$ (Rasmussen and Williams, 2008, page 89).

Axis anisotropy

A stationary axis anisotropic covariance function behaves differently along each axis in the input space. The behavior can be achieved by scaling each dimension with a parameter, i.e setting $r^2(t, t') = (t-t')^T M(t-t')$ where M is a diagonal matrix (Murphy, 2012, page 520).

4.3.3 Choice for GP emulator

The choice of covariance function for the Gaussian process is the most important part when constructing a GP emulator (Rasmussen and Williams, 2008, page 79). The covariance function should try to encode the prior assumptions of the behavior and variability of the unknown function.(Rasmussen and Williams, 2008, page 79). Given the vast number of potential covariance functions making an optimal choice is almost impossible. Instead one, or a few, of the "standard" covariance functions are commonly used, see Rasmussen and Williams (2008, page 94) for a list. Among these the Matérn covariance function is a popular and flexible choice and it will be described below.

4.3.4 Matérn covariance function

If $|t - t'|$ denotes the euclidean distance between two points the Matérn covariance function is defined as (Matérn, 1960; Rasmussen and Williams, 2008, page 84):

$$k_{\text{Matérn}}(t, t') = \sigma_f^2 \frac{2^{1-\upsilon}}{\Gamma(\upsilon)} \left(\frac{\sqrt{2\upsilon}|t - t'|}{l} \right)^{\upsilon} K_\upsilon \left(\frac{\sqrt{2\upsilon}|t - t'|}{l} \right) \tag{5}$$

where Γ is the standard gamma function and K_υ is a modified Bessel function of second kind (Roberts et al., 2012). The parameter l is called the characteristic length scale, or sometimes range, and describes the distance needed in input space to make the function values uncorrelated (Rasmussen and Williams, 2008, page 106). The length scale also describes how quickly the function values can change. σ_f^2 is called the signal variance and controls how much the function can vary from its mean value. The parameter υ determines the differentiability of of the covariance function (Roberts et al., 2012). Therefore υ controls how smooth the resulting GP is. If υ is chosen to be a half-integer ($\upsilon = p + 1/2, p = 1, 2, 3, 4....$) the expression for the Matérn covariance function simplifies to a polynomial multiplied by an exponential. As $\upsilon \to \infty$ the Matérn covariance function becomes:

$$k_{\upsilon \to \infty}(t, t') = e^{-\frac{|t-t'|^2}{2l^2}} \tag{6}$$

This special case of the Matérn covariance function is infinitely differentiable and also known as the squared exponential. The squared exponential is probably the most commonly used covariance function for emulators (Rasmussen and Williams, 2008, page 83). However the infinitely differentiability may give a too smooth representation of the actual physical behavior (Rasmussen and Williams, 2008, page 83). For the same reason the exponential covariance, $k_{\upsilon=1/2}$, which is common in spatial statistics (Gelfand, 2010, page 37), might be too rough (Rasmussen and Williams, 2008, page 85). According to Rasmussen and Williams (2008, page 85) the most interesting cases for machine learning are $k_{\upsilon=3/2}$ and $k_{\upsilon=5/2}$.

4.4 Estimate the parameters

The chosen mean and covariance function often include parameters that have to be estimated. If the observed data contains noise, the noise variance, σ_n^2 is included among the parameters. The parameters are estimated using maximum likelihood, i.e to maximize the likelihood of the observations.

$$arg \max_{\mu,\theta} p(y) \tag{7}$$

Where y is the observations. If $y \sim N(\mu, K_y)$ the expression in (7) becomes (Murphy, 2012, page 521):

$$arg \max_{\mu,\theta} \left(\frac{1}{2\pi^{n/2}|K_y|^{1/2}} e^{-\frac{1}{2}(y-\mu)^T K_y^{-1}(y-\mu)} \right) \tag{8}$$

where $|K_y|$ denotes the determinant of K_y. Assuming a linear mean, the optimal can be solved analytically using generalized least squares:

$$\hat{\mu} = \vec{x}\hat{\beta} = \vec{x}\left(\vec{x}^T K_y^{-1}\vec{x}\right)^{-1}\vec{x}^T K_y^{-1} y \tag{9}$$

For any parameter θ, substituting (9) into (8) gives the profile likelihood $p(y)$:

$$\begin{gathered} p(y) = \frac{1}{2\pi^{n/2}|K_y|^{1/2}} e^{-\frac{1}{2}y^T P y}, \\ P = K_y^{-1} - K_y^{-1}\vec{x}\left(\vec{x}^T K_y^{-1}\vec{x}\right)^{-1}\vec{x}^T K_y^{-1} \end{gathered} \tag{10}$$

Maximizing this w.r.t θ gives ML estimates of θ. The optimization can be aided by calculating analytic derivatives (see appendix, A.1).

4.5 Estimate new points given data and parameters

Assume that an unknown function, $f(\cdot)$, should be modeled by a Gaussian process. Further assume that all prior knowledge of $f(\cdot)$ is incorporated in a mean value function, $\mu(\cdot)$ and a covariance function, $k(t, t')$, with known or estimated parameters and that a set of n known points $s = \{x_i, f(x_i)|i = 1...n\}$

is available. The goal of Gaussian process prediction is to provide estimates of $f(\cdot)$ at new points $x_\star$ given estimated parameters and the set s. The predictions will use the covariance function to determine how similar the predictions should be to the data in s (Rasmussen and Williams, 2008, page 79). Given these assumptions the joint multivariate Gaussian distribution for $f(x)$ and $f(x_\star)$ is:

$$\begin{bmatrix} f(\vec{x}) \\ f(\vec{x}_\star) \end{bmatrix} \sim N\left(\begin{bmatrix} \mu(\vec{x}) \\ \mu(\vec{x}_\star) \end{bmatrix}, \begin{bmatrix} K_{xx} & K_{xx_\star} \\ K_{x_\star x} & K_{x_\star x_\star} \end{bmatrix}\right) \tag{11}$$

Where $\vec{x} = \{x_i | i = 1...n\}$, $\vec{x}_\star = \{x_\star\}$ and K_{ab} is the covariance matrix containing all pairs of points between the vectors $\vec{a}$ and $\vec{b}$. Calculating the posterior distribution $p(f(\vec{x}_\star)|f(\vec{x}))$ from the joint distribution of $f(\vec{x})$ and $f(\vec{x}_\star)$ can be done by conditioning the joint Gaussian prior distribution on the observations (Rasmussen and Williams, 2008, page 16). The resulting predictions for $p(f(\vec{x}_\star)|f(\vec{x}))$ are now given by:

$$p(f(\vec{x}_\star)|f(\vec{x})) \sim N(\mu(\vec{x}_\star) + K_{x_\star x} K_{xx}^{-1}(f(\vec{x}) - \mu(\vec{x})), K_{\star\star} - K_{x_\star x} K_{xx}^{-1} K_{xx_\star}) \tag{12}$$

Hence this distribution gives the estimated posterior mean and variance for $f(x_\star)$. If the observed data contains noise the algorithm used for prediction can be modified accordingly. If observations containing noise variance σ_n^2 are denoted $y(\cdot)$ the joint prior distribution can be written as (Rasmussen and Williams, 2008, page 16):

$$\begin{bmatrix} y(\vec{x}) \\ f(\vec{x}_\star) \end{bmatrix} \sim N\left(\begin{bmatrix} \mu(\vec{x}) \\ \mu(\vec{x}_\star) \end{bmatrix}, \begin{bmatrix} K_{xx} + \sigma_n^2 I & K_{xx_\star} \\ K_{x_\star x} & K_{x_\star x_\star} \end{bmatrix}\right) \tag{13}$$

And the posterior distribution becomes (Rasmussen and Williams, 2008, page 16):

$$\begin{aligned} p(f(\vec{x}_\star)|y(\vec{x})) \sim N(\mu(\vec{x}_\star) + K_{x_\star x}[K_{xx} + \sigma_n^2 I]^{-1}(y(\vec{x}) - \mu(\vec{x})), \\ K_{\star\star} - K_{x_\star x}[K_{xx} + \sigma_n^2 I] K_{xx_\star}) \end{aligned} \tag{14}$$

Computational aspects of the algorithm are given in appendix A.2.

4.6 Evaluation

General methods for evaluating a system in computer science or engineering often involves two steps; verification and validation (Bastos and O'Hagan, 2009). The verification is the process of determining if the system is built according to specification and that it represents the developers conceptual description of the model (Bastos and O'Hagan, 2009). During verification the developers should ask "are we creating the the system right?"(Balci, 2010). Common techniques for verification often involves developing and running different tests on the actual system (Balci, 2010). The second step is validation which is the process of determining to what degree the system manages to represent its intended usage(Bastos and O'Hagan, 2009). During validation the developers should ask "are we creating the right system?"(Balci, 2010).

4.6.1 Validation of Gaussian process model used in regression

The accuracy of the predictions given by a Gaussian process can be evaluated using a variety of different methods. A common approach is to study different residuals (Bastos and O'Hagan, 2009). Both the marginal residual which is the difference between the observed values and the values returned by the model and the conditional residuals which is the error between the predictive values for observed values not used to build the model could be used (Bastos and O'Hagan, 2009).

k-fold cross-validation
k-fold cross-validation is a technique for calculating the conditional residuals. The technique divides the original data into k disjoint sets of equal size and then training is performed on all data except for one of these k sets which is used for validation. This procedure is then repeated k times with a different validation set each time. This allows more data to be used for training and that all data points in the original data set are used for validation (Rasmussen and Williams, 2008, page 111). The number of different sets, k, can be chosen to be any number but is typically between 3 and 10 (Rasmussen and Williams, 2008, page 111). To ensure that maximum amount of data is used for training k could be chosen to be equal to the number of data points in the original data set. This special case is called leave-one-out cross-validation and it can, because of its computational cost, be difficult to apply in practice (Rasmussen and Williams, 2008, page 111).

4.6.2 Possible problems

Even if the Gaussian process is a very flexible class it can sometimes give poor predictions (Bastos and O'Hagan, 2009). Large residuals from cross validation can usually be explained by at least one of two basic reasons. The first reason is that the assumed mean and covariance functions used to train the Gaussian process are inappropriate (Bastos and O'Hagan, 2009). The second basic reason is that the parameters of the model are poorly estimated (Bastos and O'Hagan, 2009). Bad estimates can be the result of an unfortunate choice of training data or that the optimization algorithm was unable to find a global maximum (Bastos and O'Hagan, 2009; Rasmussen and Williams, 2008, page 116).

Another indication of problems with the Gaussian process is if it displays a systematic bias in its predictions. This bias often indicates an inappropriate mean structure or bad parameter estimates (Bastos and O'Hagan, 2009). The confidence interval returned by the Gaussian process could also indicate problems. For example a too wide or too narrow interval can indicate inappropriate covariance structure such as non-stationary of the underlying process or again that the parameters of the covariance function have bad estimates (Bastos and O'Hagan, 2009).

In addition to numerical evaluations graphical tools can be used for model evaluation:

Individual errors vs the Gaussian process predictions

This plot can show if there are regions that are systematically to high or to low indicating an incorrect mean function (Bastos and O'Hagan, 2009). The plot can also display if the errors are heteroscedastic or not. If the individual errors are heteroscedastic it could indicate that the actual process is non-stationary (Bastos and O'Hagan, 2009).

Standardized errors vs inputs

This plot can be used to see if errors are different across the input space, indicating that the actual process isn't stationary (Bastos and O'Hagan, 2009). The plot should display a horizontal band containing all the errors (Bastos and O'Hagan, 2009). If the plot displays a clear pattern it may indicate incorrect mean function (Bastos and O'Hagan, 2009).

Quantile quantile plot

Can be used to see if the normality assumption of the model holds. If the normality assumption holds the standardized errors should be distributed with a student-t distribution (Bastos and O'Hagan, 2009). If the points lie close to the line $y = x$ the normality assumption is reasonable (Bastos and O'Hagan, 2009). If the points cluster to a line with a different slope it may indicate that the variance of the errors were over or underestimated (Bastos and O'Hagan, 2009). Outliers at the endpoints of the QQ plot indicates non-stationarity or local fitting problems (Bastos and O'Hagan, 2009).

4.6.3 Comparing two models with each other

The performance of a Gaussian process can be evaluated using a variety of methods, see Bastos and O'Hagan (2009) for a comprehensive summary of different techniques. This thesis will use the cross validated root mean squared error, RMSE, to determine which emulator that is most suitable. The root mean squared error is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(r_k)^2} \tag{15}$$

Where r_k is the residual between the observed and estimated value, calculated using cross validation and n is the number of points. This thesis will also use the cross validated mean absolute error, MAE defined as:

$$MAE = \frac{1}{n}\sum_{k=1}^{n}|r_k| \tag{16}$$

The confidence intervals returned by the emulator will be compared using the mean absolute interval, $|I|$, defined as:

$$|I| = \frac{1}{n}\sum_{k=1}^{n}|i_k| \tag{17}$$

where i_k is the length of the confidence interval for point k.

4.7 The gpml package

The actual emulators were implemented in MATLAB using the gpml package[5]. Many alternative environments and packages exists[6]. The gpml package was chosen since it is extensive and written by one of the authors of ”Gaussian processes for machine learning” (Rasmussen and Williams, 2008), making it easy to map functions to this reference.

[5] Available at: `http://www.gaussianprocess.org/gpml/code/matlab/doc/index.html`

[6] A list of available packages can be found at: `http://www.gaussianprocess.org/#code`

5 Implementation

5.1 Reducing input data

When all available data was gathered from LPJ-GUESS it became obvious that some sort of reduction or aggregation was needed before it could be used for training a Gaussian process emulator. The main reasons being that the amount of training points needed increases with the number of inputs and that the computations needed for creating the emulator increases as $O(n^3)$. Training an emulator on all climate data would also be difficult in practice since the amount of available climate data is huge implying large memory usage. Each of the three time-series for each location were therefore reduced by calculating the mean, min, max and range values. Table 2 describes information about the annual mean value of these time-series. To reduce table size the min, max and range values where omitted from the table since they had lower correlation to the annual yield compared to the mean.

Variable	Min	Max	Correlation (Pearson's r)
Precipitation mean ($\bar{P}$)	< 0.00001	0.00018	0.05
Radiation mean ($\bar{R}$)	108	263	-0.55
Temperature mean ($\bar{T}$)	270	302	-0.50

Table 2: Describing the aggregated data for all 196 locations. The table also displays the correlation between the aggregated input variable and the output variable spring wheat. The mean for each location was calculated as the mean value of the time-series (31 years).

Since spring wheat is cultivated during the spring and summer period of the year it seemed reasonable to assume that some of the input data could be removed. This could also explain the low correlation values seen in table: 2. The optimal plant date for spring wheat differs for different locations of the world. Both plant date and harvest date for a location where calculated by LPJ-GUESS and included in the output. These outputs were used to remove all data from the inputs except 120 days before harvest. 120 days was chosen since it represented 4 month of growth and was close to the average number of growing days for spring wheat. Information about the aggregated data for these 120 days can be seen in table 3. The table values are collected from the average over the 31 years for each location.

Variable	Min	Max	Correlation (Pearson's r)
Precipitation mean ($\bar{P}_{120}$)	< 0.00001	0.00013	0.26
Radiation mean ($\bar{R}_{120}$)	125	300	-0.14
Temperature mean ($\bar{T}_{120}$)	262	304	-0.19

Table 3: Describes the aggregated data during 120 days before harvest for all 196 locations. The table also displays the correlation between the aggregated input variable and the output variable spring wheat. The mean for each location was calculated as the mean value of the time-series (31 years), only including the data 120 days before harvest.

The dependence between the lon, lat -coordinates and the annual yield was also studied, see table 4.

Variable	Correlation (Pearson's r)
Longitude	-0.25
Latitude	0.19

Table 4: Describes the correlation between coordinates and average output of spring wheat.

Each aggregated variable was plotted using histograms and analyzed to see their distributions. These plots indicated that some variables may benefit from a transformation, especially the input variable precipitation since its distribution was very skew. Both the logarithm and the cubic root transformation[7] was considered.

5.2 Model describing LPJ-GUESS

The mathematical structure of LPJ-GUESS was not investigated, instead the simulator was treated as a "black box". The problem to accurately model LPJ-GUESS using a Gaussian process approach was split into two parts. 1) Model the average yield at location i given average input, i.e climate, at that location, $\bar{c}_i$:

$$\hat{\bar{y}}_i = \gamma(\bar{c}_i) \tag{18}$$

2) Model the yield at a specific year, t, for a location given the average yield at that location and the input anomalies, i.e weather, during that year $(c_{it} - \bar{c}_i)$.

$$\hat{y}_{it} = \hat{\bar{y}}_i + \eta(c_{it} - \bar{c}_i) \tag{19}$$

where γ and η are Gaussian process emulators. The following subsections will describe the implementation of each of these parts.

[7] The transformation was considered since it has been used to effectively model rainfall (Fu et al., 2009).

5.3 Modeling average yield at location

Several emulators where constructed using different combinations of the inputs from table 2, table 3 and table 4. Each emulator used the Matérn covariance function with $\upsilon = 3/2$. Both isotropic and axis-anisotropic versions of Matérn covariance function was tested. Different mean functions of the emulator was also implemented and tested. Since the emulator only had 196 training points leave-one-out-cross validation was used for comparing and evaluating each emulator. The emulator performance was measured using RMSE.

Table 5 contains some of the created emulators and their corresponding RMSE.

Inputs	Mean Function	Covariance Function	RMSE
$\bar{P}, \bar{T}, \bar{R}$	Constant	$\underset{isotropic}{\text{Matérn}_{\upsilon=3/2}}$	0.66
$\bar{P}, \bar{T}, \bar{R}$	Constant	$\underset{anisotropic}{\text{Matérn}_{\upsilon=3/2}}$	0.65
$\bar{P}, \bar{T}, \bar{R}$	Linear	$\underset{anisotropic}{\text{Matérn}_{\upsilon=3/2}}$	0.64
$\bar{P}, \bar{T}, \bar{R}, lon, lat$	Constant	$\underset{anisotropic}{\text{Matérn}_{\upsilon=3/2}}$	0.57
$\bar{P}_{120}, \bar{T}_{120}, \bar{R}_{120}$	Constant	$\underset{anisotropic}{\text{Matérn}_{\upsilon=3/2}}$	0.72
$\bar{P}_{120}, \bar{T}_{120}, \bar{R}$	Constant	$\underset{anisotropic}{\text{Matérn}_{\upsilon=3/2}}$	0.58
$log(\bar{P}_{120}), \bar{T}_{120}, \bar{R}$	Constant	$\underset{anisotropic}{\text{Matérn}_{\upsilon=3/2}}$	0.44
$log(\bar{P}_{120}), \bar{T}_{120}, \bar{R}, lat$	Constant	$\underset{anisotropic}{\text{Matérn}_{\upsilon=3/2}}$	0.40
$log(\bar{P}_{120}), \bar{R}$	Constant	$\underset{anisotropic}{\text{Matérn}_{\upsilon=3/2}}$	0.53

Table 5: Describing a subset of all created emulators for predicting the average yield at a location.

The emulator that gave the smallest RMSE when predicting the average yield at a location was:

$$\gamma = \text{Constant} + GP(0, \underset{anisotropic}{\text{Matérn}_{\upsilon=3/2}}(\bar{c_i})) + \epsilon \tag{20}$$

Where $\bar{c_i} = [log(\bar{P}_{120}), \bar{T}_{120}, \bar{R}, lat]$. The emulator had $RMSE = 0.40, MAE = 0.27$ and $|I| = 1.59$. The estimated parameter values for this emulator can be seen in table 6 where σ_f^2 is the signal variance and σ_n^2 is the noise variance. The parameters l_i describes the diagonal element of matrix M, i.e. characteristic length scale for input i, used for modeling the axis anisotropy, see section 4.3.2. The reason why the table contains min, max and mean values for the parameters is that the optimal parameters for the emulator are different depending on the training and validation data.

Parameter	Min	Mean	Max
Constant	0.93	1.02	1.16
$e^{\sigma_f^2}$	-0.16	-0.10	-0.06
$e^{l_{log(\bar{P})}}$	0.35	0.50	0.60
$e^{l_{\bar{T}}}$	2.35	2.46	2.55
$e^{l_{\bar{R}}}$	4.62	4.70	4.88
$e^{l_{lat}}$	3.06	3.39	3.55
$e^{\sigma_n^2}$	-1.24	-1.15	-1.13

Table 6: Parameter estimates for the emulator described in equation (20)

The emulator predictions and their corresponding ±2 standard deviation interval are depicted together with the actual output for each of these 196 locations in figure 3. The predictions have been sorted to increase interpretation. Figure 4 displays the observed values vs the predicted outputs from this GP emulator. Figure 5 depicts the residuals vs the Gaussian process predictions. Figure 6 displays a QQ-plot of the residuals. Figure $7-10$ contains plot of the standardized errors vs inputs for the GP.

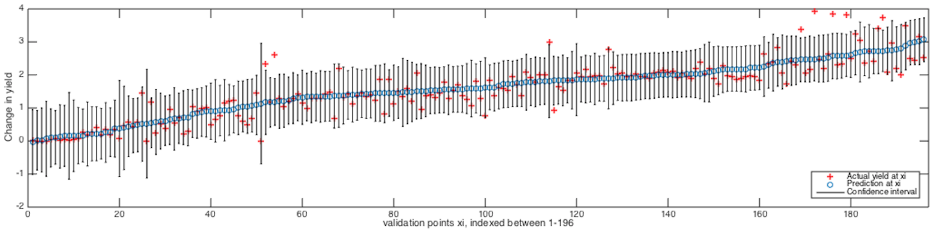

Figure 3: Emulator predictions, uncertainties and corresponding observed outputs for each of the 196 locations.

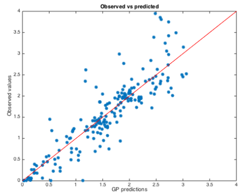

Figure 4:
Observed values vs the predicted outputs from the GP emulator. The line indicates a perfect fit.

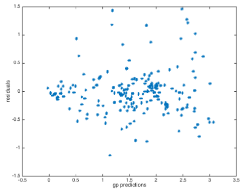

Figure 5:
Residuals vs Gaussian process predictions.

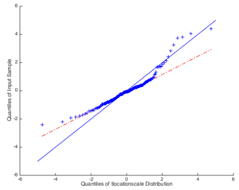

Figure 6:
QQ plot for the residuals. The blue line is $y = x$.

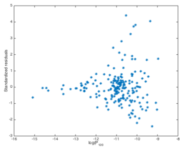

Figure 7:
Standardized residuals
vs $log(\bar{P}_{120})$

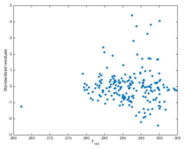

Figure 8:
Standardized residuals vs $\bar{T}_{120}$

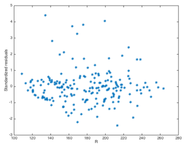

Figure 9:
Standardized residuals vs $\bar{R}$

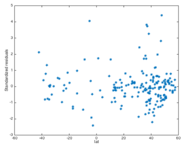

Figure 10:
Standardized residuals vs lattitude

5.3.1 Reference model

As a baseline model the mean yield was also modeled using multiple linear regression with the same covariates as the best emulator ($\bar{c}_i = [log(\bar{P}_{120}), \bar{T}_{120}, \bar{R}, lat]$). All covariates showed significant p-values and the regression had a $RMSE = 0.62, MAE = 0.44$ and $|I| = 1.59$, calculated using in-sample validation. The in-sample predictions are plotted together with the actual yield for each location in figure 11.

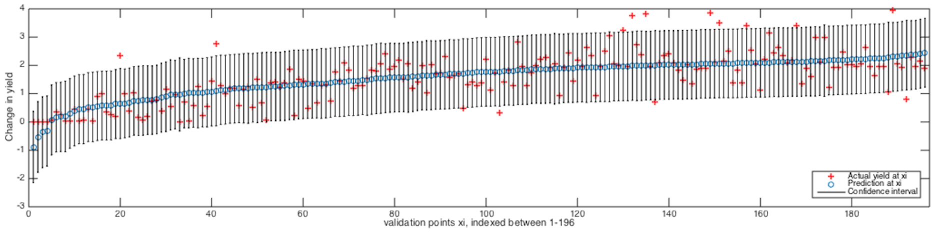

Figure 11: Multiple linear regression predictions, uncertainties and corresponding observed outputs for each of the 196 locations.

5.4 Model change in yield for a specific year

Several emulators where constructed using annual anomalies for the yearly data and the anomalies for the data collected 120 days before harvest. All the anomalies where calculated as the input for that year minus the average input over all 31 years at that specific location. Table 7 contains information of all the calculated anomalies.

Anomaly	Min	Mean	Max
Yield	-1.7	≈ 0	7.1
$\bar{P}$	$-3.5 \cdot 10^{-5}$	≈ 0	$5.6 \cdot 10^{-5}$
$\bar{R}$	-56	≈ 0	48
$\bar{T}$	-4.6	≈ 0	7.8
$\bar{P}_{120}$	$-5.1 \cdot 10^{-5}$	≈ 0	$8.0 \cdot 10^{-5}$
$\bar{R}_{120}$	-55	≈ 0	42
$\bar{T}_{120}$	-5.6	≈ 0	6.3

Table 7: Describes the calculated anomalies.

Again each emulator used the Matérn covariance function with $\upsilon = 3/2$ for creating the covariance matrix. Both isotropic and anisotropic versions of Matérn covariance function was tested. Different mean functions of the emulator were also implemented and tested. Since the emulator had $196 \cdot 31 = 6076$ training points k-fold cross validation with k = 7 was used for comparing and evaluating each emulator. The emulator performance was measured using RMSE. Table 8 contains some of the created emulators and their corresponding RMSE.

Inputs	Mean Function	Covariance Function	RMSE
$\bar{P}, \bar{T}, \bar{R}$	Constant	$\text{Matérn}_{\upsilon=3/2}$ *isotropic*	0.411
$\bar{P}, \bar{T}, \bar{R}$	Constant	$\text{Matérn}_{\upsilon=3/2}$ *anisotropic*	0.409
$\bar{P}, \bar{T}, \bar{R}$	Linear	$\text{Matérn}_{\upsilon=3/2}$ *anisotropic*	0.409
$\bar{P}, \bar{T}, \bar{R}, lon, lat$	Constant	$\text{Matérn}_{\upsilon=3/2}$ *anisotropic*	0.404
$\bar{P}, \bar{T}, \bar{R}, lat$	Constant	$\text{Matérn}_{\upsilon=3/2}$ *anisotropic*	0.406
$\bar{P}_{120}, \bar{T}_{120}, \bar{R}$	Constant	$\text{Matérn}_{\upsilon=3/2}$ *anisotropic*	0.415
$\bar{P}_{120}, \bar{T}_{120}, \bar{R}_{120}$	Constant	$\text{Matérn}_{\upsilon=3/2}$ *anisotropic*	0.415

Table 8: Describing a subset of all created emulators for predicting the change in yield during a specific year.

The emulator that gave the smallest RMSE when predicting the average change in yield during a specific year at a location was:

$$\gamma = \text{Constant} + GP(0, \underset{anisotropic}{\text{Matérn}_{\upsilon=3/2}}(\bar{c_i})) + \epsilon \tag{21}$$

Where $\bar{c_i} = [\bar{P}, \bar{T}, \bar{R}, lon, lat]$ and $\epsilon \sim N(0, \sigma_n^2)$. The emulator had $RMSE = 0.404, MAE = 0.22$ and $|I| = 1.60$. The estimated parameter values for this emulator can be seen in table: 9.

Parameter	Min	Mean	Max
Constant	0.01	0.10	0.23
$e^{\sigma_f^2}$	-2.14	-1.24	-0.50
$e^{l_{\bar{P}}}$	$4.58 \cdot 10^{-5}$	$4.59 \cdot 10^{-5}$	$4.60 \cdot 10^{-5}$
$e^{l_{\bar{T}}}$	-1.12	-0.36	0.48
$e^{l_{\bar{R}}}$	0.074	2.17	3.21
$e^{l_{lon}}$	1.55	3.92	5.22
$e^{l_{lat}}$	1.36	2.68	3.64
$e^{\sigma_n^2}$	-1.01	-0.95	-0.92

Table 9: Parameter estimates for the emulator described in equation (21).

The predictions from the emulator with lowest RMSE are plotted together with a ±2 standard deviation interval in figure 12. Figure 13 displays the observed values vs the predicted outputs from this GP emulator. Figure 14 depicts the residuals vs the Gaussian process predictions. Figure 15 displays a QQ-plot of the residuals. Figure $16 - 20$ contains plots of standardized errors vs inputs for the emulator.

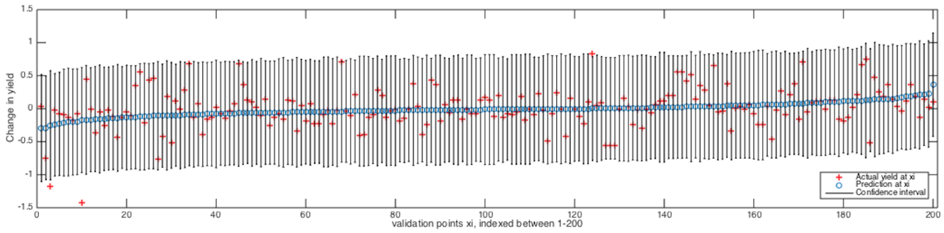

Figure 12: Emulator predictions and corresponding outputs of change in annual yield of spring wheat. Only a subset of the total number of points is plotted.

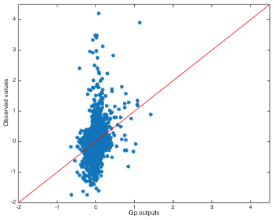

Figure 13:
Observed values vs the predicted outputs from the GP emulator. The line indicates a perfect fit.

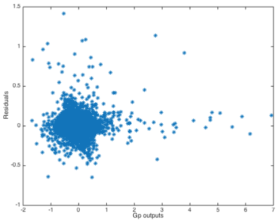

Figure 14:
Residuals vs the Gaussian process predictions

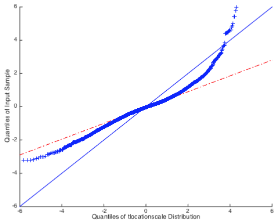

Figure 15:
QQ plot for the residuals. The blue line is $y = x$.

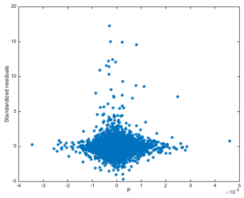

Figure 16:
Standardized residuals vs $\bar{P}$.

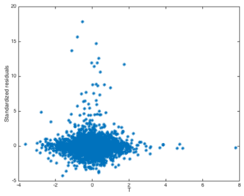

Figure 17:
Standardized residuals vs $\bar{T}$.

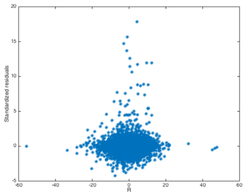

Figure 18:
Standardized residuals vs $\bar{R}$.

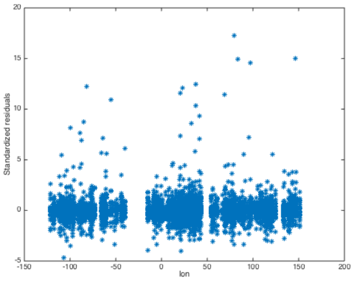

Figure 19:
Standardized residuals vs longitude.

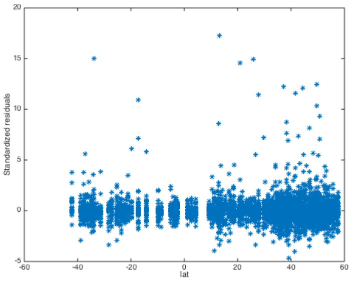

Figure 20:
Standardized residuals vs latitude.

5.4.1 Model each location independently

As a test the change in yield for a specific location at a given year was also modeled by constructing a GP emulator for each location. This was done to investigate if the predictions could be improved by assuming that each locations behaved differently to changes in input space. Each GP emulator were constructed using the same structure and covariates (except longitude and latitude) as the model in (21). Hence 196 different emulators were constructed. Each of these emulators were then evaluated by excluding one of the 31 years before training and then using this year for validation. Since the performance measurements depends on which years that are randomly excluded during training and validation the approach was done 10 times and then the average of all these runs was calculated to ensure a reliable values. The approach gave an average RMSE=0.483, MAE= 0.24 and $|I| = 1.26$. The predictions for one of these runs are plotted together with the actual change in yield for each location in figure 21, note the difference in scale on the y-axis from figure 12. The emulator prediction and the corresponding observations are plotted in figure 22

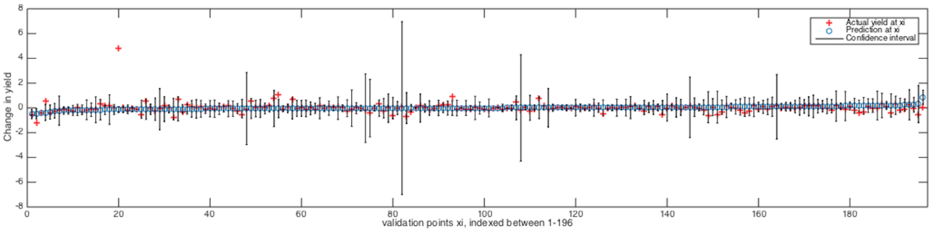

Figure 21: Emulator predictions and corresponding outputs of change in annual harvest. Each location is predicted independently.

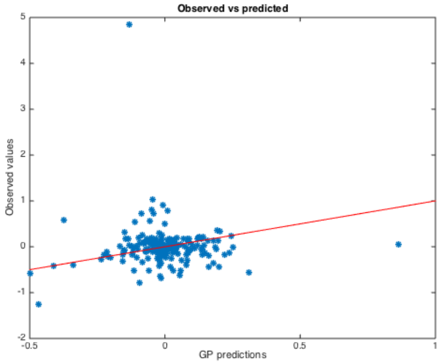

Figure 22: Emulator predictions and corresponding observations. Each location is predicted independently. The red line indicates a perfect fit.

6 Results and analysis

6.1 Emulator for predicting the average yield

The emulator for predicting the average yield at a specific location managed to capture some of the points and performed better than multiple linear regression. The estimated signal variance, i.e. variance in $k(t, t')$, of the underlying function was larger than the estimated noise variance, i.e. σ_n^2, indicating that the emulator manages to capture most of the variance in the response variable.

As seen in figure 5 and figure $7 - 10$ the emulator had heteroscedastic errors. The QQ-plot indicated that the estimated variance was too high with heteroscedastic errors mainly for large values, consistent with the model's inability to capture large yields. These results indicate that the underlying process might be non-stationary. Figure 7 displays heteroscedasity due to precipitation, indicating that the non-stationary behavior might depend on this input. The choice of using a stationary covariance function to model the behavior may therefore be insufficient. The locations corresponding to the highest residuals can be seen in figure 23. The plot shows that the emulator typically gives poor predictions for a few places in north America and at remote islands. However, predictions for nearby points are often good, making it hard to draw consistent conclusions.

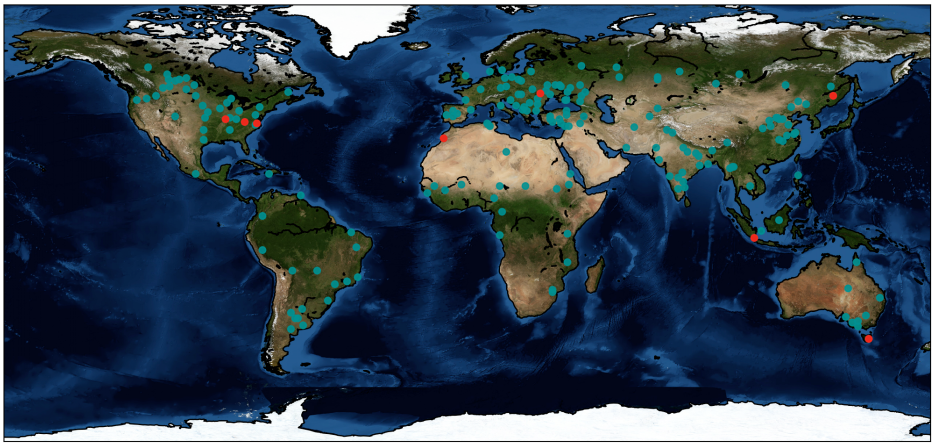

Figure 23: Locations where the emulator had large residuals are plotted in red.

6.2 Emulator for predicting the change in yield for a specific year

The emulator used for predicting the change in yield from the average yield at different locations was unable to capture much of the underlying structure as seen in figure 12 and 13. The estimated parameters also indicates poor performance since the noise variance was larger than the signal variance, indicating that most of the variability is "explained" by noise. The emulator had heteroscedastic errors, again most severely so for large yields. The QQ-plot indicated that the estimated variance was too high since the points clustered to a line with slope less than 1. The performance of the emulator might be explained by the skewness of the response variable or that different changes in the inputs affects the harvest differently at different locations around the world. However no particular improvement in RMSE were achieved when a GP emulator was constructed for each of the 196 locations. Since both approaches gave poor estimates the reason might be poor choice of covariates. It is not unrealistic that important information is lost by building the emulator on aggregated data and that the yearly change in yield depends on changes in daily measurements instead of changes in mean values during 365 days for the different inputs. A more extensive research of the yearly behavior of LPJ-GUESS for each location is needed to discover these behaviors.

7 Conclusion

Several Gaussian process emulators where constructed and as seen in figure 4 and 13 the results was far from satisfactory. However the emulator used for modeling the average yield at a location indicated that the Gaussian process approach may work if researched further. Possibly by using a more complex covariance function and by researching different aggregations of the inputs further.

The created emulators for modeling the change in yield for a specific year was unable of LPJ-GUESS. The performance measurements for these predictions didn't increase much whether an emulator was constructed at each location or if it was constructed for all locations at once. This may imply that poor choices of covariates was used for training the emulators.

7.1 Discussion

The behavior of LPJ-GUESS was much more complex to model than anticipated. To be able to model LPJ-GUESS a more extensive analysis on how to aggregate the inputs and a more complex covariance structure than presented in this thesis is likely needed. A starting point would be to investigate the heteroscedasticity due to precipitation, see figure 7, and try to incorporate this in the emulator. Even if the results from this thesis are unsatisfactory it can be seen as a starting-point for further and related research.

While reading the report it might seem incorrect to only try to model LPJ-GUESS using a Gaussian processes with the Matérn covariance function since this decision was done before any research or evaluations was performed. However the work-load of this master thesis would have been too large to perform within the time-frame if multiple emulator strategies and different covariance functions would be implemented and evaluated. The decision to use a Gaussian process approach as the emulator strategy is although not entirely unsupported. Gaussian processes are very flexible and often used as a first choice for emulators.

Worth noting is that the implemented emulators in this thesis uses the data 120 days before harvest as input for a new location. The exact harvest-date however is not known before LPJ-GUESS is executed which implies that this needs to be predicted as well for the emulator to be able to predict new points.

8 Future work

This thesis only covers a small subset of all the possible solutions for modeling the behaviors of LPJ-GUESS. Future research includes trying a different emulator approach from section 1.6 or research and implement a more complex Gaussian process emulator. Techniques that could increase the performance of the emulator includes opening LPJ-GUESS and perform an extensive analysis of its internal mathematical structure and researching the inputs more throughly. The performance can also be increased by collecting more training data from LPJ-GUESS and by configuring it to execute grid cells with inputs related to high uncertainty within the emulator. A more formal method for choosing and comparing different mean value and covariance functions could also be developed to increase the chances of finding a better emulator.

Another thing that could increase the performance might be to research techniques to implement an emulator for each grid cell.

A Appendix

A.1 Maximizing the profile likelihood

Since the expression for the profile likelihood contains an exponential function the logarithm of the profile likelihood is often used. Assuming a zero mean, $\mu = 0$, the derivatives of the log profile likelihood w.r.t each parameter can be calculated as (Rasmussen and Williams, 2008, page 114):

$$\frac{\partial}{\partial \theta_j} log(p(y|\mu)) = \frac{1}{2} y^T K_y^{-1} \frac{\partial K_y}{\partial \theta_j} K_y^{-1} y - \frac{1}{2} tr \left(K_y^{-1} \frac{\partial K_y}{\partial \theta_j} \right) \tag{22}$$

The inverse of the matrix K_y takes $O(n^3)$ to compute and the gradient takes $O(n^2)$ per parameter to compute (Murphy, 2012, page 521). When estimating θ by maximizing the log profile likelihood in this manner there is no guarantee that the the global maximum is found. This is because the marginal likelihood may suffer from multiple local optimas (Rasmussen and Williams, 2008, page 115).

A.2 Computer considerations. Estimation of new points given data and parameters

The main step when predicting mean and variance of an unknown point $f(\vec{x}_\star)$ is the calculation of K^{-1}. See section 4.5.
Since the covariance matrix K is designed to be a positive definite matrix it can be represented as a product of two matrices using the Cholesky decomposition:

$$A = LL^T \tag{23}$$

Where L is a lower triangular matrix (Rasmussen and Williams, 2008, page 202). The decomposition offers a more numerical stable way to solving linear equations on the form $Ax = b$ by first solving the triangular system $Ly = b$ by forward substitution and then the triangular system $L^T x = y$ by back substitution. The decomposition can also be used to calculate the inverse of a matrix since (Murphy, 2012, page 524):

$$A^{-1} = (L^T)^{-1} L^{-1} \tag{24}$$

The Cholesky decomposition takes $n^3/6 = O(n^3)$ operations to compute for an $n \times n$ matrix. Both the forward and backward substitution steps require $n^2/2 = O(n^2)$ operations. (Rasmussen and Williams, 2008, page 202). The actual algorithm for performing the calculations can be seen in (Murphy, 2012, page 524). The Cholesky decomposition can be used to calculate the determinant of a positive definite symmetric matrix since (Rasmussen and Williams, 2008, page 203):

$$|A| = \prod_{i=1}^{n} L_{ii}^2 \tag{25}$$

A.3 Figures

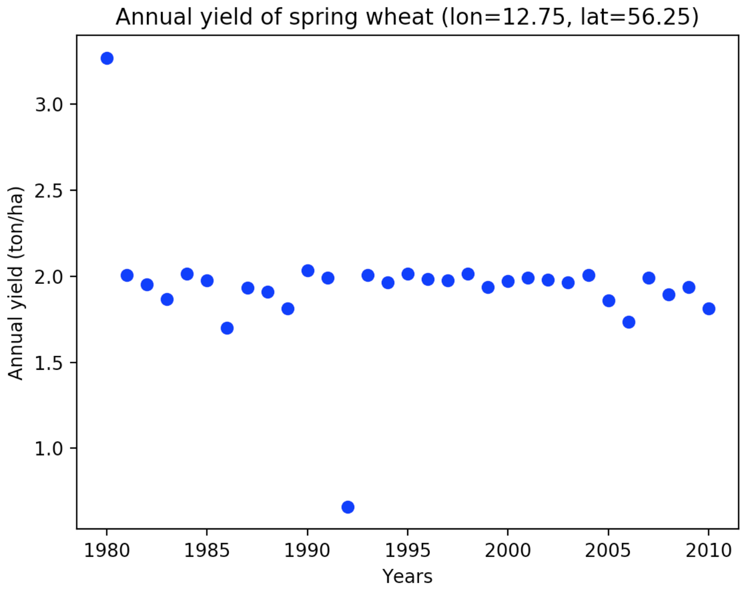

Figure 24: Annual yield of spring wheat at lon=12.75, lat=56.25. This location is close to Lund.

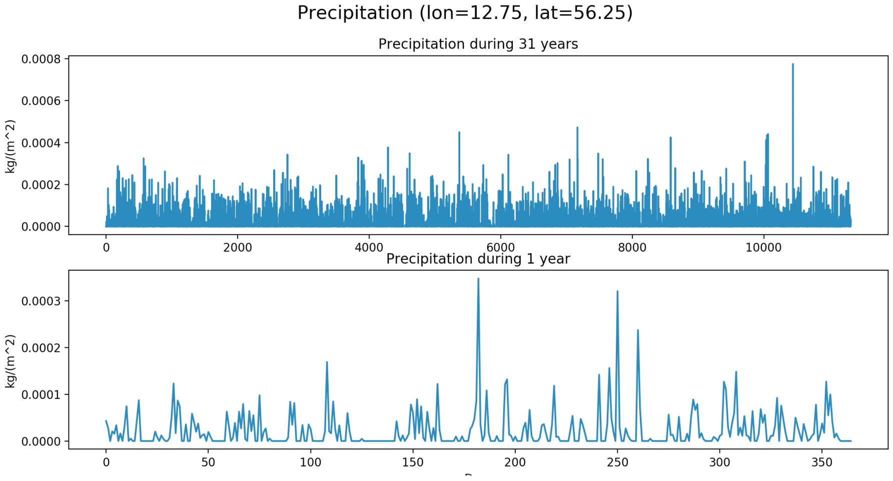

Figure 25: Plot of precipitation between 1980 and 2010 at lon=12.75, lat=56.25. The bottom panel displays precipitation from the year 2000.

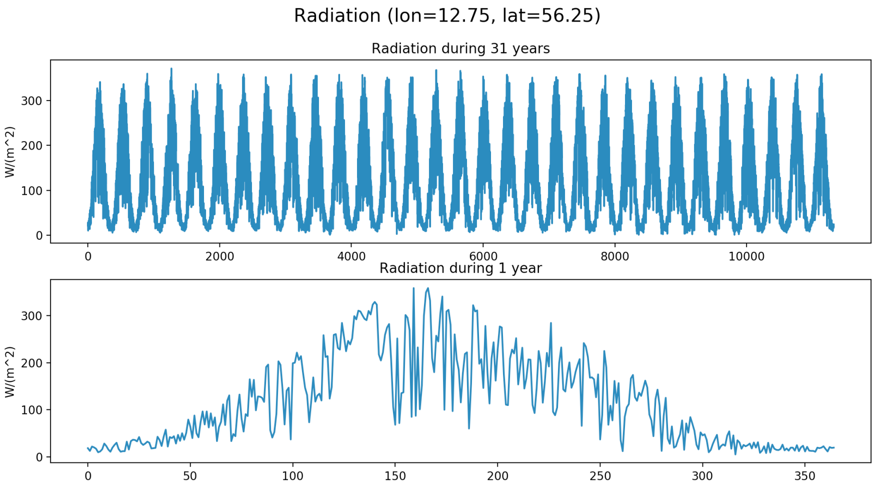

Figure 26: Plot of radiation between 1980 and 2010 at lon=12.75, lat=56.25. The bottom panel displays radiation from the year 2000.

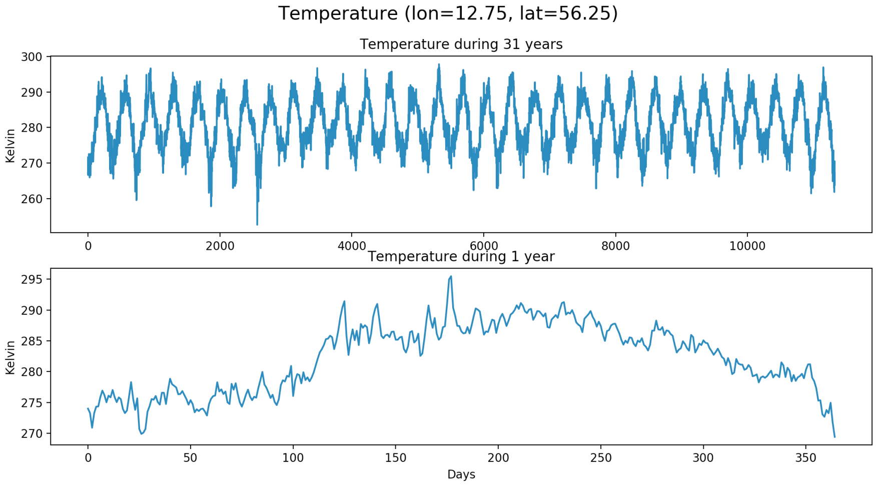

Figure 27: Plot of temperature between 1980 and 2010 at lon=12.75, lat=56.25. The bottom panel displays temperature from the year 2000.

References

AgMIP. Agricultural model intercomparison and improvement project. 2017. URL `http://www.agmip.org/about/`.

O. Balci. Golden rules of verification, validation, testing, and certification of modeling and simulation applications. 2010.

L. S. Bastos and A. O'Hagan. Diagnostics for gaussian process emulators. *Technometrics*, 51(4):425–438, 2009. doi: 10.1198/tech.2009.08019.

S. Castruccio, D. J. Mcinerney, M. L. Stein, F. L. Crouch, R. L. Jacob, and E. J. Moyer. Statistical emulation of climate model projections based on precomputed gcm runs*. *Journal of Climate*, 27(5):1829–1844, 2014. doi: 10.1175/jcli-d-13-00099.1.

G. Fu, N. R. Viney, and S. P. Charles. Evaluation of various root transformations of daily precipitation amounts fitted with a normal distribution for australia. *Theoretical and Applied Climatology*, 99(1-2):229–238, May 2009. doi: 10.1007/s00704-009-0137-6.

B. Geiger, C. Meurey, D. Lajas, L. Franchistéguy, D. Carrer, and J.-L. Roujean. Near real-time provision of downwelling shortwave radiation estimates derived from satellite observations. *Meteorological Applications*, 15(3), 2008. doi: 10.1002/met.84.

A. E. Gelfand. *Handbook of spatial statistics.* CRC Press, 2010.

C. Hillel, Daniel. Rosenzweig. *Handbook of climate change and agroecosystems.* Imperial College Pr., 2015.

M. C. Kennedy and A. Ohagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001. doi: 10.1111/1467-9868.00294.

S. Klassen and B. a. Bugbee. Shortwave radiation. *Micrometeorology in Agricultural Systems Agronomy Monograph*, 2005. doi: 10.2134/agronmonogr47.c3.

G. Lindgren, R. Holger, and M. Sandsten. *Stationary stochastic processes for scientists and engineers.* CRC Press, 2014.

B. Matérn. *Spatial variation. Stochastic models and their application to some problems in forest surveys and other sampling investigations.* PhD thesis, Stockholm University, Stockholm, Sweden, June 1960.

Merriam-Webster. Spring wheat, 2017. URL `https://www.merriam-webster.com/dictionary/springwheat`.

K. P. Murphy. *Machine learning: a probabilistic perspective.* The MIT Press, 2012.

NC-University. Temperature relation to agriculture, 2010. URL `http://climate.ncsu.edu/edu/k12/temperature/ag`.

NC-University. Precipitation types, 2013. URL `http://climate.ncsu.edu/edu/k12/.PrecipTypes`.

A. O'Hagan. Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering & System Safety*, (10-11), 2006. doi: 10.1016/j.ress.2005.11.025.

A. O'Hagan, S. Conti, J. P. Gosling, and J. E. a. Oakley. Gaussian process emulation of dynamic computer codes. *Biometrika*, 96(3):663–676, 2009. doi: 10.1093/biomet/asp028.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2008.

S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for timeseries modelling. 2012.

B. Smith, I. C. Prentice, and M. T. Sykes. Representation of vegetation dynamics in the modelling of terrestrial ecosystems: comparing two contrasting approaches within european climate space. *Global Ecology and Biogeography*, 10:621–637, 2001. doi: 10.1046/j.1466-822x.2001.t01-1-00256.x.

H. Wackernagel. *Multivariate geostatistics: an introduction with applications*. Springer, 2010.

R. P. Wilson, Andrew Gordon. Adams. Gaussian process kernels for pattern discovery and extrapolation. 2013.

D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996. doi: 10.1162/neco.1996.8.7.1341.

MATHEMATICAL STATISTICS | LUND UNIVERSITY

MASTER THESIS Emulators for dynamic vegetation models
STUDENT Olof Olsson
SUPERVISOR Johan Lindström
EXAMINER Andreas Jakobsson

Predicting the predictions

POPULAR SCIENCE SUMMARY **Olof Olsson**

The increase in world population along with observed and expected changes in the environment due to human actions implies risks that future food production will be insufficient. These risks can be minimized by predicting these changes using complex computer programs called simulators. These simulators are usually computationally expensive, implying that predictions where many different future climate scenarios are considered will be expensive. This thesis investigates a method for modeling the behavior of the simulator using a machine learning technique.

The ability to understand and predict the natural behaviors surroundings us has been one of mankind's biggest challenges. We are now able to predict phenomenons such as the weather quite accurately and the models are constantly improving. One of the things we are trying to predict is the future impact different climate changes have on the environment in which we live. Knowing how the environment will behave in the future is essential for proactively respond to these changes.

Predicting the environment is difficult and the models usually involves simulating the real world processes day by day until it reaches the state it should predict. The predictions from these simulators can therefore take many hours to obtain, even for a modern supercomputer. The simulator would be of much more practical use if the predictions could be obtained much faster. What if the behavior of the simulator could be predicted just like the behavior of the real world phenomenon could be predicted by the simulator? It would be a model trying to predict a model that tries to predict the real world. This is exactly what this thesis has tried to implement.

By using recent advances in machine learning and computer science it can be possible for a computer program to ”learn” how the simulator behaves in different situations. The aim of the thesis was to ”learn” the behavior of a simulator used to predict future production of spring wheat at different locations around the world when it was affected by different climate changes.

The problem of accurately modeling the simulator was split into two parts. The first part was to model the average yield at each location given average climate input at that location. The second part was to model the yield at a specific year for a location given the average yield at that location and the climate input anomalies during that year.

Modeling the simulator was difficult in practice and the results was far from perfect, however they managed to show that the used strategy might be useful for predicting some quantities within the simulator. With more time and further research about the internal structure of the simulator and the different simulator inputs the model might be improved so it could be used for practical applications.

One may say that the current version of the model is of little use but at the same time the models used for predicting the weather is far from perfect but we still use and rely on them everyday.

Master’s Theses in Mathematical Sciences 2017:E66
ISSN 1404-6342

LUTFMS-3336-2017

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/