Learning non-Gaussian Time Series using the Box-Cox Gaussian Process

Gonzalo Rios and Felipe Tobar

Department of Mathematical Engineering
Center for Mathematical Modelling
Universidad de Chile

December 18, 2018

In a nutshell

- Gaussian process for time series
- A recipe to construct non-Gaussian processes
- The proposed Box-Cox Gaussian process

In a nutshell

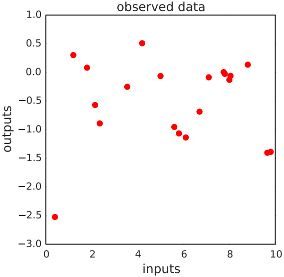
- Gaussian process for time series
- A recipe to construct non-Gaussian processes
- The proposed Box-Cox Gaussian process

In a nutshell

- Gaussian process for time series
- A recipe to construct non-Gaussian processes
- The proposed Box-Cox Gaussian process

Motivation

The Regression Problem



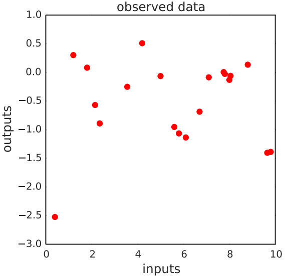
Definition.

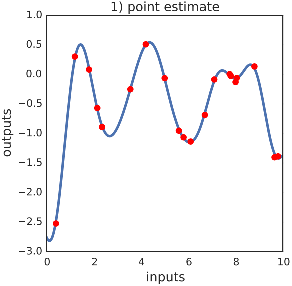
A generative model is a joint probability distribution over all variables of interest.

- Interpolate and extrapolate
- Probabilistic estimation
- Statistics and samples

Motivation

The Regression Problem





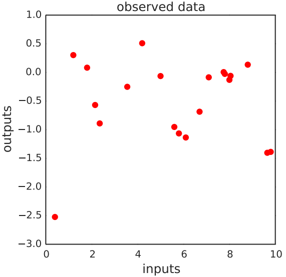
Definition.

A generative model is a joint probability distribution over all variables of interest.

- Interpolate and extrapolate
- Probabilistic estimation
- Statistics and samples

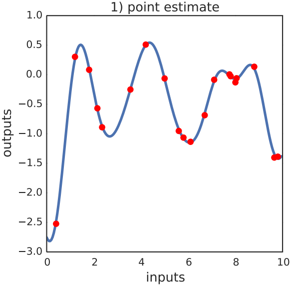
Motivation

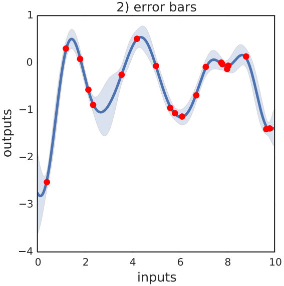
The Regression Problem



Definition.
A generative model is a joint probability distribution over all variables of interest.

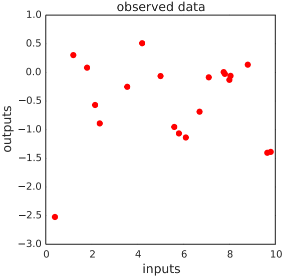
- Interpolate and extrapolate
- Probabilistic estimation
- Statistics and samples





Motivation

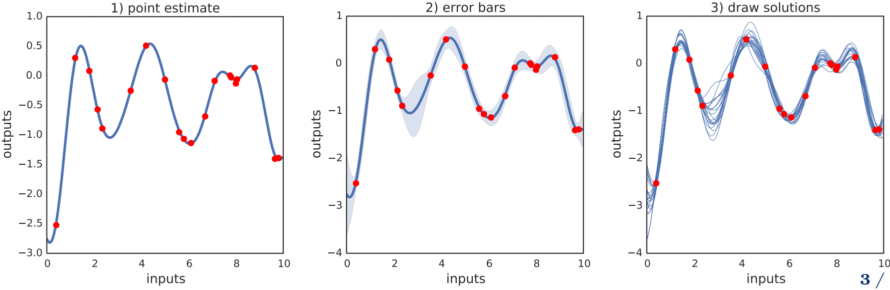
The Regression Problem



Definition.

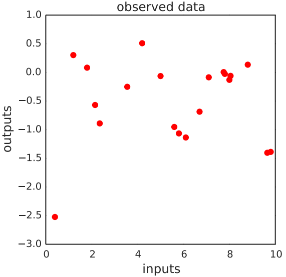
A generative model is a joint probability distribution over all variables of interest.

- Interpolate and extrapolate
- Probabilistic estimation
- Statistics and samples



Motivation

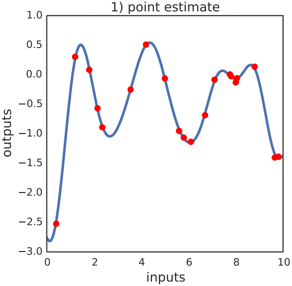
The Regression Problem

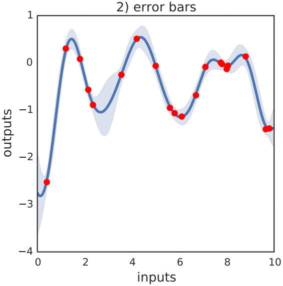


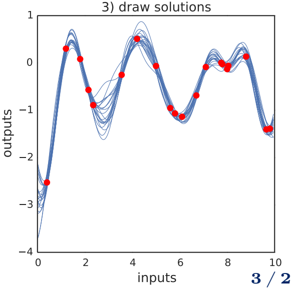
Definition.

A generative model is a joint probability distribution over all variables of interest.

- Interpolate and extrapolate
- Probabilistic estimation
- Statistics and samples





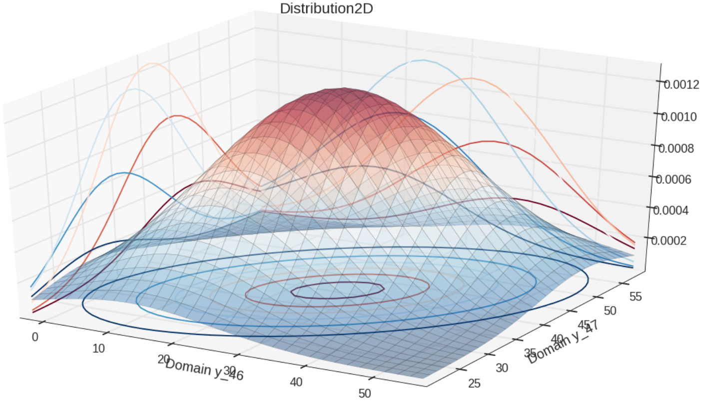


Gaussian Processes

Multivariate Normal Distribution

A random vector $y \in \mathbb{R}^n$ is said to follow a normal distribution with mean $\mu \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ if its density function is

$$\boldsymbol{\mathcal{N}_n\left(y; \mu, \Sigma\right) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\Sigma\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(y-\mu)^\top \Sigma^{-1}(y-\mu)}}$$



Gaussian Processes

Generative Model for Time Series

Definition.

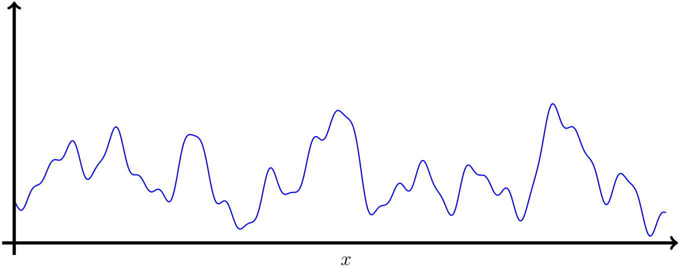
A Gaussian process (denoted GP) is a stochastic process such that any finite collection of values follows a multivariate normal distribution.

Gaussian Processes

Generative Model for Time Series

Definition.

A Gaussian process (denoted GP) is a stochastic process such that any finite collection of values follows a multivariate normal distribution.

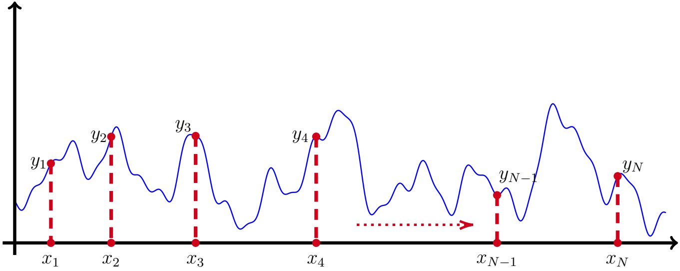


Gaussian Processes

Generative Model for Time Series

Definition.

A Gaussian process (denoted GP) is a stochastic process such that any finite collection of values follows a multivariate normal distribution.

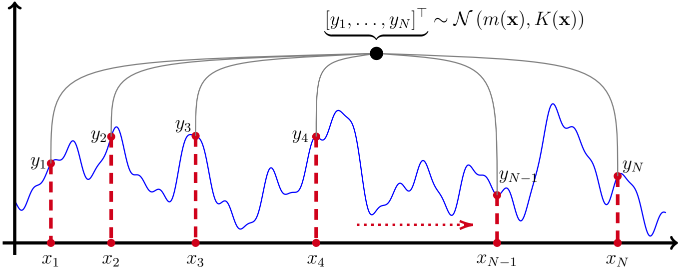


Gaussian Processes

Generative Model for Time Series

Definition.

A Gaussian process (denoted GP) is a stochastic process such that any finite collection of values follows a multivariate normal distribution.

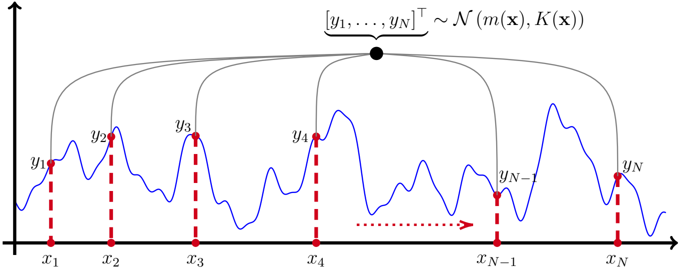


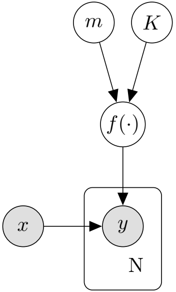
Gaussian Processes

Generative Model for Time Series

Definition.

A Gaussian process (denoted GP) is a stochastic process such that any finite collection of values follows a multivariate normal distribution.



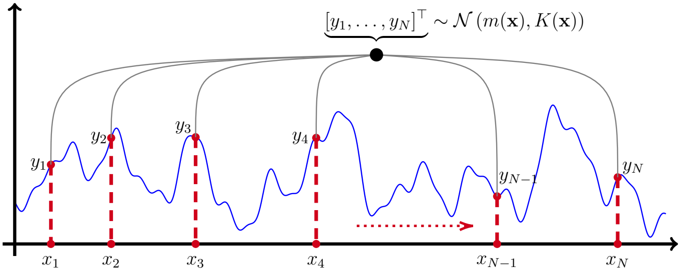


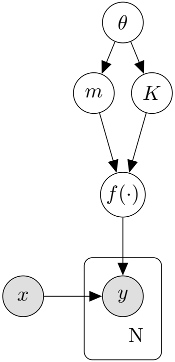
Gaussian Processes

Generative Model for Time Series

Definition.

A Gaussian process (denoted GP) is a stochastic process such that any finite collection of values follows a multivariate normal distribution.





Gaussian Processes

Prior Distribution over Functions

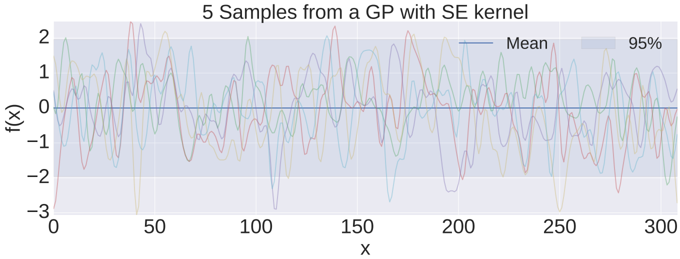
A **GP** is a *prior* distribution over functions, denoted as

$$\boldsymbol{f}(\boldsymbol{x}) \sim \mathcal{GP}\left(\boldsymbol{m}(\boldsymbol{x}), \boldsymbol{k}(\boldsymbol{x}, \bar{\boldsymbol{x}})\right),$$

and it is fully-determined by a mean function $\boldsymbol{m}(\cdot)$ and a covariance kernel $\boldsymbol{k}(\cdot,\cdot)$. The *de-facto* kernel is the *Squared Exponential*

$$\boldsymbol{k_{SE}}(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \boldsymbol{\sigma}^2 \exp\left(-\frac{(\boldsymbol{x}-\bar{\boldsymbol{x}})^2}{\boldsymbol{l}^2}\right),$$

where $\boldsymbol{\sigma}^2 > \mathbf{0}, \boldsymbol{l} > \mathbf{0}$ are the *hyperparameters*.



Gaussian Processes

A Posteriori Distribution

By observing data, we can calculate the *posterior distribution*:

- Update the model
- Point predictions
- Confidence intervals
- Sample functions

Gaussian Processes

A Posteriori Distribution

By observing data, we can calculate the *posterior distribution*:

- Update the model
- Point predictions
- Confidence intervals
- Sample functions

Gaussian Processes

A Posteriori Distribution

By observing data, we can calculate the *posterior distribution*:

- Update the model
- Point predictions
- Confidence intervals
- Sample functions

Gaussian Processes

A Posteriori Distribution

By observing data, we can calculate the *posterior distribution*:

- Update the model
- Point predictions
- Confidence intervals
- Sample functions

Gaussian Processes

A Posteriori Distribution

By observing data, we can calculate the *posterior distribution*:

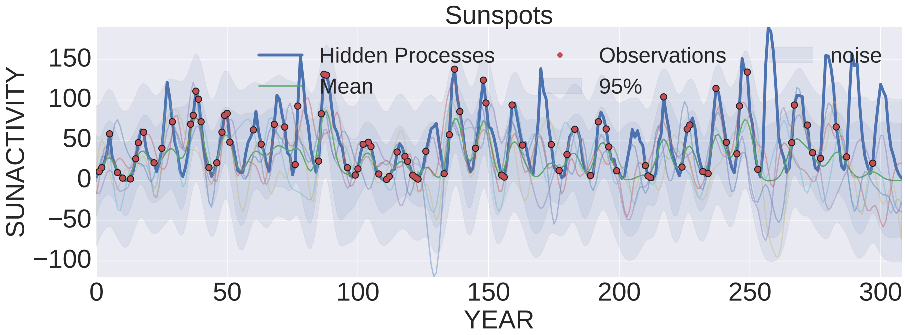
- Update the model
- Point predictions
- Confidence intervals
- Sample functions

Gaussian Processes

A Posteriori Distribution

By observing data, we can calculate the *posterior distribution*:

- Update the model
- Point predictions
- Confidence intervals
- Sample functions



Gaussian Processes

Learning Hyperparameters

To learn the *hyperparameters* of a **GP**

- Maximize likelihood
- Minimize negative log-likelihood

$$-\log \mathcal{L}(\theta) = \frac{n}{2} \log (2\pi) + \frac{1}{2} \log |K_\theta| + \frac{1}{2} (y - m(x))^\top K_\theta^{-1} (y - m(x))$$

  - Quasi-Newton BFGS method (gradient)
  - Powell's method (derivative-free)
  - Markov Chain Monte Carlo methods (sampling)

Gaussian Processes

Learning Hyperparameters

To learn the *hyperparameters* of a **GP**

- Maximize likelihood
- Minimize negative log-likelihood

$$-\log \mathcal{L}(\theta) = \frac{n}{2}\log(2\pi) + \frac{1}{2}\log|K_\theta| + \frac{1}{2}(y - m(x))^\top K_\theta^{-1}(y - m(x))$$

  - Quasi-Newton BFGS method (gradient)
  - Powell's method (derivative-free)
  - Markov Chain Monte Carlo methods (sampling)

Gaussian Processes

Learning Hyperparameters

To learn the *hyperparameters* of a **GP**

- Maximize likelihood
- Minimize negative log-likelihood

$$-\log \mathcal{L}(\theta) = \frac{n}{2} \log (2\pi) + \frac{1}{2} \log |K_{\theta}| + \frac{1}{2} (y - m(x))^{\top} K_{\theta}^{-1} (y - m(x))$$

  - Quasi-Newton BFGS method (gradient)
  - Powell's method (derivative-free)
  - Markov Chain Monte Carlo methods (sampling)

Gaussian Processes

Learning Hyperparameters

To learn the *hyperparameters* of a **GP**

- Maximize likelihood
- Minimize negative log-likelihood

$$-\log \mathcal{L}(\theta) = \frac{n}{2}\log(2\pi) + \frac{1}{2}\log|K_{\theta}| + \frac{1}{2}(y - m(x))^{\top} K_{\theta}^{-1}(y - m(x))$$

  - Quasi-Newton BFGS method (gradient)
  - Powell's method (derivative-free)
  - Markov Chain Monte Carlo methods (sampling)

Gaussian Processes

Learning Hyperparameters

To learn the *hyperparameters* of a **GP**

- Maximize likelihood
- Minimize negative log-likelihood

$$-\log \mathcal{L}(\theta) = \frac{n}{2} \log(2\pi) + \frac{1}{2} \log |K_{\theta}| + \frac{1}{2} (y - m(x))^{\top} K_{\theta}^{-1} (y - m(x))$$

  - Quasi-Newton BFGS method (gradient)
  - Powell's method (derivative-free)
  - Markov Chain Monte Carlo methods (sampling)

Gaussian Processes

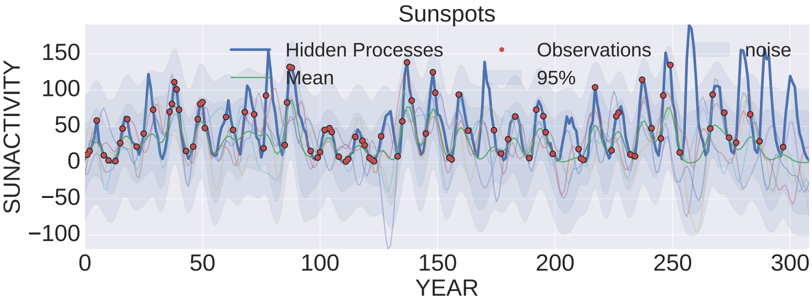
Learning Hyperparameters

To learn the *hyperparameters* of a **GP**

- Maximize likelihood
- Minimize negative log-likelihood

$$-\log \mathcal{L}(\theta) = \frac{n}{2} \log(2\pi) + \frac{1}{2} \log |K_\theta| + \frac{1}{2} (y - m(x))^\top K_\theta^{-1} (y - m(x))$$

  - Quasi-Newton BFGS method (gradient)
  - Powell's method (derivative-free)
  - Markov Chain Monte Carlo methods (sampling)



Gaussian Processes

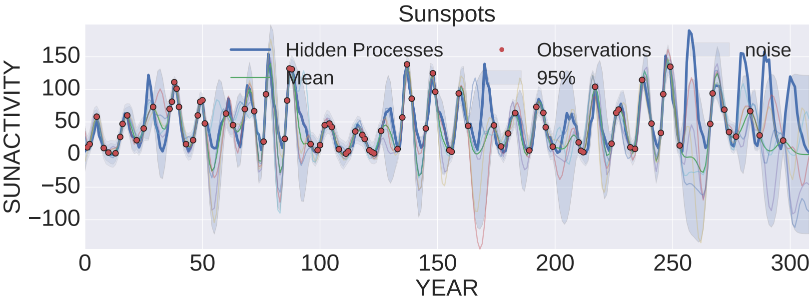
Learning Hyperparameters

To learn the *hyperparameters* of a **GP**

- Maximize likelihood
- Minimize negative log-likelihood

$$-\log \mathcal{L}(\theta) = \frac{n}{2}\log(2\pi) + \frac{1}{2}\log|K_{\boldsymbol{\theta}}| + \frac{1}{2}(y - m(x))^\top K_{\boldsymbol{\theta}}^{-1}(y - m(x))$$

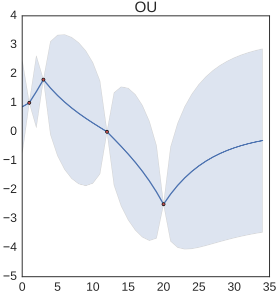
  - Quasi-Newton BFGS method (gradient)
  - Powell's method (derivative-free)
  - Markov Chain Monte Carlo methods (sampling)



Gaussian Processes

The Kernel Choice

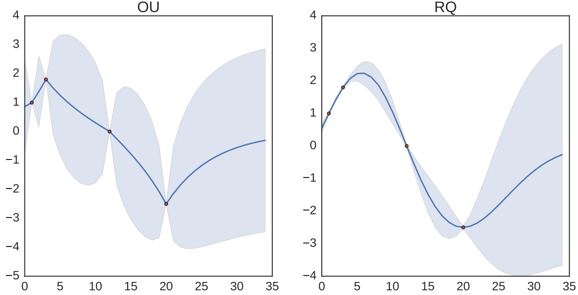
- Ornstein-Uhlenbeck: $\boldsymbol{k_{OU}\left(x, \bar{x}\right) = \sigma^2 \exp\left(-\frac{|x-\bar{x}|}{2l^2}\right)}$
- Rational Quadratic: $\boldsymbol{k_{RQ}\left(x, \bar{x}\right) = \sigma^2 \left(1 + \frac{|x-\bar{x}|^2}{2\alpha l^2}\right)^{-\alpha}}$
- Spectral Mixture:
  $\boldsymbol{k_{SM}\left(x, \bar{x}\right) = \sigma^2 \exp\left(-\frac{|x-\bar{x}|^2}{2l^2}\right) \cos\left(\frac{2\pi}{p}|x-\bar{x}|\right)}$



Gaussian Processes

The Kernel Choice

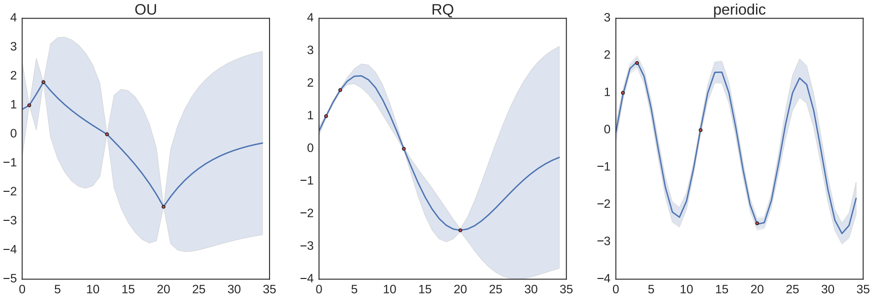
- Ornstein-Uhlenbeck: $k_{OU}(x, \bar{x}) = \sigma^2 \exp\left(-\frac{|x-\bar{x}|}{2l^2}\right)$
- Rational Quadratic: $k_{RQ}(x, \bar{x}) = \sigma^2 \left(1 + \frac{|x-\bar{x}|^2}{2\alpha l^2}\right)^{-\alpha}$
- Spectral Mixture:
  $k_{SM}(x, \bar{x}) = \sigma^2 \exp\left(-\frac{|x-\bar{x}|^2}{2l^2}\right) \cos\left(\frac{2\pi}{p}|x-\bar{x}|\right)$



Gaussian Processes

The Kernel Choice

- Ornstein-Uhlenbeck: $\boldsymbol{k_{OU}\left(x, \bar{x}\right) = \sigma^2 \exp\left(-\frac{|x-\bar{x}|}{2l^2}\right)}$
- Rational Quadratic: $\boldsymbol{k_{RQ}\left(x, \bar{x}\right) = \sigma^2 \left(1 + \frac{|x-\bar{x}|^2}{2\alpha l^2}\right)^{-\alpha}}$
- Spectral Mixture:
  $\boldsymbol{k_{SM}\left(x, \bar{x}\right) = \sigma^2 \exp\left(-\frac{|x-\bar{x}|^2}{2l^2}\right) \cos\left(\frac{2\pi}{p}|x-\bar{x}|\right)}$



Gaussian Processes

Weaknesses

GP is a useful modelling tool

- Closed-form formulas for training
- Closed-form formulas for prediction

But, the *hypothesis* that the observations are jointly normally distributed does not always hold in practice

- Non-Gaussian noise
- Asymmetric density
- Bounded domain
- Heavy tails

e.g. observations positive/bounded by a physical/economic restriction.

Gaussian Processes

Weaknesses

GP is a useful modelling tool

- Closed-form formulas for training
- Closed-form formulas for prediction

But, the *hypothesis* that the observations are jointly normally distributed does not always hold in practice

- Non-Gaussian noise
- Asymmetric density
- Bounded domain
- Heavy tails

e.g. observations positive/bounded by a physical/economic restriction.

Gaussian Processes

Weaknesses

GP is a useful modelling tool

- Closed-form formulas for training
- Closed-form formulas for prediction

But, the *hypothesis* that the observations are jointly normally distributed does not always hold in practice

- Non-Gaussian noise
- Asymmetric density
- Bounded domain
- Heavy tails

e.g. observations positive/bounded by a physical/economic restriction.

Gaussian Processes

Weaknesses

GP is a useful modelling tool

- Closed-form formulas for training
- Closed-form formulas for prediction

But, the *hypothesis* that the observations are jointly normally distributed does not always hold in practice

- Non-Gaussian noise
- Asymmetric density
- Bounded domain
- Heavy tails

e.g. observations positive/bounded by a physical/economic restriction.

Gaussian Processes

Weaknesses

GP is a useful modelling tool

- Closed-form formulas for training
- Closed-form formulas for prediction

But, the *hypothesis* that the observations are jointly normally distributed does not always hold in practice

- Non-Gaussian noise
- Asymmetric density
- Bounded domain
- Heavy tails

e.g. observations positive/bounded by a physical/economic restriction.

Gaussian Processes

Weaknesses

GP is a useful modelling tool

- Closed-form formulas for training
- Closed-form formulas for prediction

But, the *hypothesis* that the observations are jointly normally distributed does not always hold in practice

- Non-Gaussian noise
- Asymmetric density
- Bounded domain
- Heavy tails

e.g. observations positive/bounded by a physical/economic restriction.

Gaussian Processes

Weaknesses

GP is a useful modelling tool

- Closed-form formulas for training
- Closed-form formulas for prediction

But, the *hypothesis* that the observations are jointly normally distributed does not always hold in practice

- Non-Gaussian noise
- Asymmetric density
- Bounded domain
- Heavy tails

e.g. observations positive/bounded by a physical/economic restriction.

Gaussian Processes

Weaknesses

GP is a useful modelling tool

- Closed-form formulas for training
- Closed-form formulas for prediction

But, the *hypothesis* that the observations are jointly normally distributed does not always hold in practice

- Non-Gaussian noise
- Asymmetric density
- Bounded domain
- Heavy tails

e.g. observations positive/bounded by a physical/economic restriction.

Gaussian Processes
Weaknesses

GP is a useful modelling tool

- Closed-form formulas for training
- Closed-form formulas for prediction

But, the *hypothesis* that the observations are jointly normally distributed does not always hold in practice

- Non-Gaussian noise
- Asymmetric density
- Bounded domain
- Heavy tails

e.g. observations positive/bounded by a physical/economic restriction.

Warped Gaussian Processes

Definition

Warped Gaussian Process (**WGP**) approach is based on:

- A latent **GP** $x_t \sim \mathcal{GP}\left(m(t), k\left(t, \bar{t}\right)\right)$
- A parametric non-linear $\mathcal{C}^1$ bijective scalar map $\varphi_{\theta} : \mathcal{Y} \rightarrow \mathcal{X}$
- Define the coordinate-wise transformation $\left[\Phi_{\theta} x\right]_t = \varphi_{\theta}^{-1}(x_t)$
- Apply Φ_{θ} to induce a new process as $y_t = \left[\Phi_{\theta} x\right]_t$
- Denoted this **WGP** as $y_t \sim \mathcal{WGP}\left(\phi_{\theta}, m\left(t\right), k\left(t, \bar{t}\right)\right)$

The induced process y_t is non-Gaussian!

Warped Gaussian Processes

Definition

Warped Gaussian Process (**WGP**) approach is based on:

- A latent **GP** $x_t \sim \mathcal{GP}\left(m(t), k\left(t, \bar{t}\right)\right)$
- A parametric non-linear $\mathcal{C}^1$ bijective scalar map $\varphi_\theta : \mathcal{Y} \rightarrow \mathcal{X}$
- Define the coordinate-wise transformation $\left[\Phi_\theta x\right]_t = \varphi_\theta^{-1}(x_t)$
- Apply Φ_θ to induce a new process as $y_t = \left[\Phi_\theta x\right]_t$
- Denoted this **WGP** as $y_t \sim \mathcal{WGP}\left(\phi_\theta, m\left(t\right), k\left(t, \bar{t}\right)\right)$

The induced process y_t is non-Gaussian!

Warped Gaussian Processes

Definition

Warped Gaussian Process (**WGP**) approach is based on:

- A latent **GP** $x_t \sim \mathcal{GP}\left(m(t), k\left(t, \bar{t}\right)\right)$
- A parametric non-linear $\mathcal{C}^1$ bijective scalar map $\varphi_{\theta} : \mathcal{Y} \to \mathcal{X}$
- Define the coordinate-wise transformation $\left[\Phi_{\theta} x\right]_t = \varphi_{\theta}^{-1}(x_t)$
- Apply Φ_{θ} to induce a new process as $y_t = \left[\Phi_{\theta} x\right]_t$
- Denoted this **WGP** as $y_t \sim \mathcal{WGP}\left(\phi_{\theta}, m\left(t\right), k\left(t, \bar{t}\right)\right)$

The induced process y_t is non-Gaussian!

Warped Gaussian Processes

Definition

Warped Gaussian Process (**WGP**) approach is based on:

- A latent **GP** $x_t \sim \mathcal{GP}\left(m(t), k\left(t, \bar{t}\right)\right)$
- A parametric non-linear $\mathcal{C}^1$ bijective scalar map $\varphi_\theta : \mathcal{Y} \to \mathcal{X}$
- Define the coordinate-wise transformation $\left[\Phi_\theta x\right]_t = \varphi_\theta^{-1}(x_t)$
- Apply Φ_θ to induce a new process as $y_t = \left[\Phi_\theta x\right]_t$
- Denoted this **WGP** as $y_t \sim \mathcal{WGP}\left(\phi_\theta, m\left(t\right), k\left(t, \bar{t}\right)\right)$

The induced process y_t is non-Gaussian!

Warped Gaussian Processes

Definition

Warped Gaussian Process (**WGP**) approach is based on:

- A latent **GP** $\boldsymbol{x_t \sim \mathcal{GP}\left(m(t), k\left(t, \bar{t}\right)\right)}$
- A parametric non-linear $\boldsymbol{\mathcal{C}^1}$ bijective scalar map $\boldsymbol{\varphi_\theta : \mathcal{Y} \to \mathcal{X}}$
- Define the coordinate-wise transformation $\boldsymbol{\left[\Phi_\theta x\right]_t = \varphi_\theta^{-1}(x_t)}$
- Apply $\boldsymbol{\Phi_\theta}$ to induce a new process as $\boldsymbol{y_t = \left[\Phi_\theta x\right]_t}$
- Denoted this **WGP** as $y_t \sim \mathcal{WGP}\left(\phi_\theta, m\left(t\right), k\left(t, \bar{t}\right)\right)$

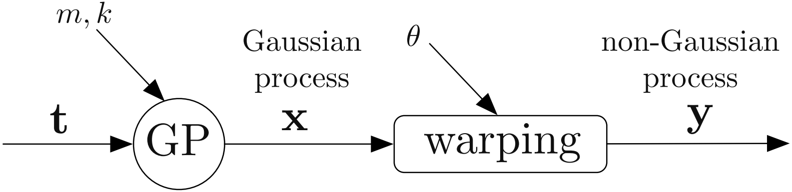
The induced process $\boldsymbol{y_t}$ is non-Gaussian!

Warped Gaussian Processes

Definition

Warped Gaussian Process (**WGP**) approach is based on:

- A latent **GP** $x_t \sim \mathcal{GP}\left(m(t), k\left(t, \bar{t}\right)\right)$
- A parametric non-linear $\mathcal{C}^1$ bijective scalar map $\varphi_\theta : \mathcal{Y} \to \mathcal{X}$
- Define the coordinate-wise transformation $[\Phi_\theta x]_t = \varphi_\theta^{-1}(x_t)$
- Apply Φ_θ to induce a new process as $y_t = [\Phi_\theta x]_t$
- Denoted this **WGP** as $y_t \sim \mathcal{WGP}\left(\phi_\theta, m\left(t\right), k\left(t, \bar{t}\right)\right)$



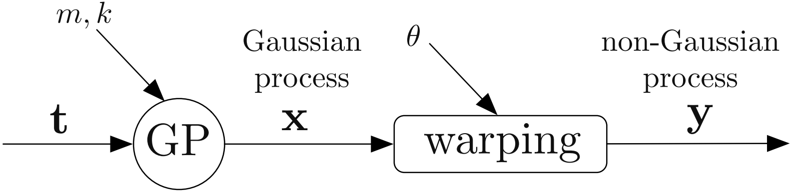
The induced process y_t is non-Gaussian!

Warped Gaussian Processes

Definition

Warped Gaussian Process (**WGP**) approach is based on:

- A latent **GP** $x_t \sim \mathcal{GP}(m(t), k(t, \bar{t}))$
- A parametric non-linear $\mathcal{C}^1$ bijective scalar map $\varphi_\theta : \mathcal{Y} \to \mathcal{X}$
- Define the coordinate-wise transformation $[\Phi_\theta x]_t = \varphi_\theta^{-1}(x_t)$
- Apply Φ_θ to induce a new process as $y_t = [\Phi_\theta x]_t$
- Denoted this **WGP** as $y_t \sim \mathcal{WGP}(\phi_\theta, m(t), k(t, \bar{t}))$



The induced process y_t is non-Gaussian!

Warped Gaussian Processes

Closed-Form Formulas

Let be $\mathbf{t} = [t_1, ..., t_n]^\top$ and $\mathbf{t}' = [t'_1, ..., t'_m]^\top$, where $\mathbf{x} \sim \mathcal{N}(\mu_\mathbf{x}, \Sigma_\mathbf{x})$ and $\mathbf{x}' \sim \mathcal{N}(\mu_{\mathbf{x}'}, \Sigma_{\mathbf{x}'})$ are the resp. finite distributions of x_t . With $\mathbf{x} = \varphi(\mathbf{y})$ and $\mathbf{x}' = \varphi(\mathbf{y}')$, through the change-of-variables theorem we have:

- **Density:** $p(\mathbf{y}) = \prod_{i=1}^{n} \frac{d\varphi(y_i)}{dy} \mathcal{N}(\varphi(\mathbf{y}) | \mu_\mathbf{x}, \Sigma_\mathbf{x})$
- **Posterior:** $p(\mathbf{y}|\mathbf{y}') = \prod_{i=1}^{n} \frac{d\varphi(y_i)}{dy} \mathcal{N}\left(\varphi(\mathbf{y}) | \mu_{\mathbf{x}|\mathbf{x}'}, \Sigma_{\mathbf{x}|\mathbf{x}'}\right)$[1]
- **NLL:** $-\log p(\mathbf{y}|\theta_x, \theta_\varphi) = \frac{n}{2}\log(2\pi) + \frac{1}{2}|K_\theta| - \sum_{i=1}^{n} \log\left(\frac{d\varphi(y_i)}{dy}\right) + \frac{1}{2}(\varphi(\mathbf{y}) - m_\theta)^\top K_\theta^{-1}(\varphi(\mathbf{y}) - m_\theta)$

WGP have closed-form formulas as **GP**!

[1] The posterior mean and covariance of $\mathbf{x}|\mathbf{x}'$ are $\mu_{\mathbf{x}|\mathbf{x}'} = \mu_\mathbf{x} + \Sigma_{\mathbf{x}\mathbf{x}'}\Sigma_{\mathbf{x}'\mathbf{x}'}^{-1}(\mathbf{x}' - \mu_{\mathbf{x}'})$ and $\Sigma_{\mathbf{x}|\mathbf{x}'} = \Sigma_{\mathbf{x}\mathbf{x}} - \Sigma_{\mathbf{x}\mathbf{x}'}\Sigma_{\mathbf{x}'\mathbf{x}'}^{-1}\Sigma_{\mathbf{x}'\mathbf{x}}$ resp.

Warped Gaussian Processes

Closed-Form Formulas

Let be $\mathbf{t} = [t_1, ..., t_n]^\top$ and $\mathbf{t}' = [t'_1, ..., t'_m]^\top$, where $\mathbf{x} \sim \mathcal{N}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}})$ and $\mathbf{x}' \sim \mathcal{N}(\mu_{\mathbf{x}'}, \Sigma_{\mathbf{x}'})$ are the resp. finite distributions of x_t . With $\mathbf{x} = \varphi(\mathbf{y})$ and $\mathbf{x}' = \varphi(\mathbf{y}')$, through the change-of-variables theorem we have:

- **Density:** $p(\mathbf{y}) = \prod_{i=1}^{n} \frac{d\varphi(y_i)}{dy} \mathcal{N}\left(\varphi(\mathbf{y}) | \mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}\right)$
- **Posterior:** $p\left(\mathbf{y}|\mathbf{y}'\right) = \prod_{i=1}^{n} \frac{d\varphi(y_i)}{dy} \mathcal{N}\left(\varphi\left(\mathbf{y}\right) | \mu_{\mathbf{x}|\mathbf{x}'}, \Sigma_{\mathbf{x}|\mathbf{x}'}\right)$[1]
- **NLL:** $-\log p(\mathbf{y}|\theta_x, \theta_\varphi) = \frac{n}{2}\log(2\pi) + \frac{1}{2}\left|K_\theta\right| - \sum_{i=1}^{n} \log\left(\frac{d\varphi(y_i)}{dy}\right) + \frac{1}{2}\left(\varphi(\mathbf{y}) - m_\theta\right)^\top K_\theta^{-1}\left(\varphi(\mathbf{y}) - m_\theta\right)$

WGP have closed-form formulas as **GP**!

[1] The posterior mean and covariance of $\mathbf{x}|\mathbf{x}'$ are $\mu_{\mathbf{x}|\mathbf{x}'} = \mu_{\mathbf{x}} + \Sigma_{\mathbf{x}\mathbf{x}'}\Sigma_{\mathbf{x}'\mathbf{x}'}^{-1}\left(\mathbf{x}' - \mu_{\mathbf{x}'}\right)$ and $\Sigma_{\mathbf{x}|\mathbf{x}'} = \Sigma_{\mathbf{x}\mathbf{x}} - \Sigma_{\mathbf{x}\mathbf{x}'}\Sigma_{\mathbf{x}'\mathbf{x}'}^{-1}\Sigma_{\mathbf{x}'\mathbf{x}}$ resp.

Warped Gaussian Processes
Closed-Form Formulas

Let be $\mathbf{t} = [t_1, ..., t_n]^\top$ and $\mathbf{t}' = [t'_1, ..., t'_m]^\top$, where $\mathbf{x} \sim \mathcal{N}(\mu_\mathbf{x}, \Sigma_\mathbf{x})$ and $\mathbf{x}' \sim \mathcal{N}(\mu_{\mathbf{x}'}, \Sigma_{\mathbf{x}'})$ are the resp. finite distributions of x_t . With $\mathbf{x} = \varphi(\mathbf{y})$ and $\mathbf{x}' = \varphi(\mathbf{y}')$, through the change-of-variables theorem we have:

- **Density:** $p(\mathbf{y}) = \prod_{i=1}^{n} \frac{d\varphi(y_i)}{dy} \mathcal{N}\left(\varphi(\mathbf{y}) | \mu_\mathbf{x}, \Sigma_\mathbf{x}\right)$
- **Posterior:** $p\left(\mathbf{y}|\mathbf{y}'\right) = \prod_{i=1}^{n} \frac{d\varphi(y_i)}{dy} \mathcal{N}\left(\varphi\left(\mathbf{y}\right) | \mu_{\mathbf{x}|\mathbf{x}'}, \Sigma_{\mathbf{x}|\mathbf{x}'}\right)$[1]
- **NLL:** $-\log p(\mathbf{y}|\theta_x, \theta_\varphi) = \frac{n}{2}\log(2\pi) + \frac{1}{2}|K_\theta| - \sum_{i=1}^{n} \log\left(\frac{d\varphi(y_i)}{dy}\right) + \frac{1}{2}\left(\varphi(\mathbf{y}) - m_\theta\right)^\top K_\theta^{-1}\left(\varphi(\mathbf{y}) - m_\theta\right)$

WGP have closed-form formulas as **GP**!

[1]The posterior mean and covariance of $\mathbf{x}|\mathbf{x}'$ are $\mu_{\mathbf{x}|\mathbf{x}'} = \mu_\mathbf{x} + \Sigma_{\mathbf{x}\mathbf{x}'}\Sigma_{\mathbf{x}'\mathbf{x}'}^{-1}\left(\mathbf{x}' - \mu_{\mathbf{x}'}\right)$ and $\Sigma_{\mathbf{x}|\mathbf{x}'} = \Sigma_{\mathbf{x}\mathbf{x}} - \Sigma_{\mathbf{x}\mathbf{x}'}\Sigma_{\mathbf{x}'\mathbf{x}'}^{-1}\Sigma_{\mathbf{x}'\mathbf{x}}$ resp.

Warped Gaussian Processes

Closed-Form Formulas

Let be $\mathbf{t} = [t_1, ..., t_n]^\top$ and $\mathbf{t'} = [t'_1, ..., t'_m]^\top$, where $\mathbf{x} \sim \mathcal{N}(\mu_\mathbf{x}, \Sigma_\mathbf{x})$ and $\mathbf{x'} \sim \mathcal{N}(\mu_{\mathbf{x'}}, \Sigma_{\mathbf{x'}})$ are the resp. finite distributions of x_t . With $\mathbf{x} = \varphi(\mathbf{y})$ and $\mathbf{x'} = \varphi(\mathbf{y'})$, through the change-of-variables theorem we have:

- **Density:** $p(\mathbf{y}) = \prod_{i=1}^{n} \frac{d\varphi(y_i)}{dy} \mathcal{N}\left(\varphi(\mathbf{y}) | \mu_\mathbf{x}, \Sigma_\mathbf{x}\right)$
- **Posterior:** $p\left(\mathbf{y}|\mathbf{y'}\right) = \prod_{i=1}^{n} \frac{d\varphi(y_i)}{dy} \mathcal{N}\left(\varphi\left(\mathbf{y}\right) | \mu_{\mathbf{x}|\mathbf{x'}}, \Sigma_{\mathbf{x}|\mathbf{x'}}\right)$[1]
- **NLL:** $-\log p(\mathbf{y}|\theta_x, \theta_\varphi) = \frac{n}{2}\log(2\pi) + \frac{1}{2}|K_\theta| - \sum_{i=1}^{n} \log\left(\frac{d\varphi(y_i)}{dy}\right) + \frac{1}{2}\left(\varphi(\mathbf{y}) - m_\theta\right)^\top K_\theta^{-1}\left(\varphi(\mathbf{y}) - m_\theta\right)$

WGP have closed-form formulas as **GP**!

[1] The posterior mean and covariance of $\mathbf{x}|\mathbf{x'}$ are $\mu_{\mathbf{x}|\mathbf{x'}} = \mu_\mathbf{x} + \Sigma_{\mathbf{x}\mathbf{x'}}\Sigma_{\mathbf{x'}\mathbf{x'}}^{-1}(\mathbf{x'} - \mu_{\mathbf{x'}})$ and $\Sigma_{\mathbf{x}|\mathbf{x'}} = \Sigma_{\mathbf{x}\mathbf{x}} - \Sigma_{\mathbf{x}\mathbf{x'}}\Sigma_{\mathbf{x'}\mathbf{x'}}^{-1}\Sigma_{\mathbf{x'}\mathbf{x}}$ resp.

Warped Gaussian Processes

Closed-Form Formulas

Let be $\mathbf{t} = [t_1, ..., t_n]^\top$ and $\mathbf{t'} = [t'_1, ..., t'_m]^\top$, where $\mathbf{x} \sim \mathcal{N}(\mu_\mathbf{x}, \Sigma_\mathbf{x})$ and $\mathbf{x'} \sim \mathcal{N}(\mu_{\mathbf{x'}}, \Sigma_{\mathbf{x'}})$ are the resp. finite distributions of x_t . With $\mathbf{x} = \varphi(\mathbf{y})$ and $\mathbf{x'} = \varphi(\mathbf{y'})$, through the change-of-variables theorem we have:

- **Density:** $p(\mathbf{y}) = \prod_{i=1}^{n} \frac{d\varphi(y_i)}{dy} \mathcal{N}\left(\varphi(\mathbf{y}) | \mu_\mathbf{x}, \Sigma_\mathbf{x}\right)$
- **Posterior:** $p\left(\mathbf{y}|\mathbf{y'}\right) = \prod_{i=1}^{n} \frac{d\varphi(y_i)}{dy} \mathcal{N}\left(\varphi\left(\mathbf{y}\right) | \mu_{\mathbf{x}|\mathbf{x'}}, \Sigma_{\mathbf{x}|\mathbf{x'}}\right)$[1]
- **NLL:** $-\log p(\mathbf{y}|\theta_x, \theta_\varphi) = \frac{n}{2}\log(2\pi) + \frac{1}{2}\left|K_\theta\right| - \sum_{i=1}^{n} \log\left(\frac{d\varphi(y_i)}{dy}\right) + \frac{1}{2}\left(\varphi(\mathbf{y}) - m_\theta\right)^\top K_\theta^{-1}\left(\varphi(\mathbf{y}) - m_\theta\right)$

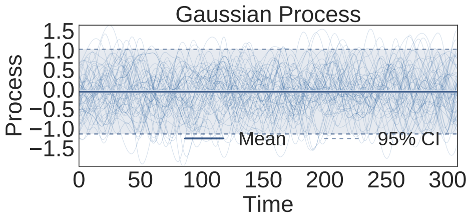
WGP have closed-form formulas as **GP**!

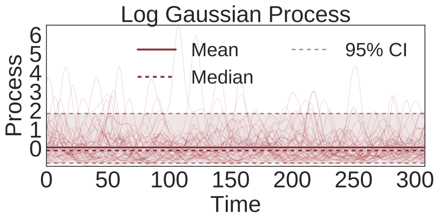
[1] The posterior mean and covariance of $\mathbf{x}|\mathbf{x'}$ are $\mu_{\mathbf{x}|\mathbf{x'}} = \mu_\mathbf{x} + \Sigma_{\mathbf{x}\mathbf{x'}}\Sigma_{\mathbf{x'}\mathbf{x'}}^{-1}\left(\mathbf{x'} - \mu_{\mathbf{x'}}\right)$ and $\Sigma_{\mathbf{x}|\mathbf{x'}} = \Sigma_{\mathbf{x}\mathbf{x}} - \Sigma_{\mathbf{x}\mathbf{x'}}\Sigma_{\mathbf{x'}\mathbf{x'}}^{-1}\Sigma_{\mathbf{x'}\mathbf{x}}$ resp.

Warped Gaussian Processes

Example: Log Gaussian Processes

- A standard strategy to transform non-Gaussian positive values is to apply the logarithmic function $\varphi_{\log}(y) = \log(y)$
- y_t is a positive-valued heavy-tailed stochastic processes (**LogGP**)
- The n-th moment of y_t is given by $\mathbb{E}_y\left[y_t^n\right] = \exp\left(nm_{x_t} + \frac{1}{2}n^2\sigma_{x_t}^2\right)$





Warped Gaussian Processes

Computation of Predictions

For any map ϕ, we can calculate explicitly

- Median: $Q_{\frac{1}{2}}(y_t) = \phi^{-1}\left(Q_{\frac{1}{2}}(x_t)\right) = \phi^{-1}(m(t))$
- Confidence intervals: $I^p_{y_t} = \left[\phi^{-1}(m(t) - z_p\sigma(t)), \phi^{-1}(m(t) + z_p\sigma(t))\right]$ [2]
- Sampling: $x(\mathbf{t}) \sim \mathcal{N}(m(\mathbf{t}), k(\mathbf{t}, \mathbf{t}))$ and then $y(\mathbf{t}) = \phi^{-1}(x(\mathbf{t}))$

The moments can be efficiently computed numerically using the Gauss-Hermite (**GH**) quadrature. The k-approx.[3] of the mean of y_t is

$$\mathbb{E}[y_t] = \int \phi^{-1}(x)\, p_{x_t}(x)\, dx \approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^{k} w_i \phi^{-1}\left(\sqrt{2}\sigma(t)x_i + m(t)\right)$$

where $m(t)$ and $\sigma(t)$ are the mean and std. dev. of the latent x_t, and the weights $\{w_i\}_{i=1}^{k}$ and locations $\{x_i\}_{i=1}^{k}$ are given by **GH** quadrature.

---

[2] $\sigma(t) = \sqrt{k(t,t)}$ and z_p is the p-quantile of standard normal (ex. $z_{0.975} \approx 1.96$)

[3] It is exact when the integrand is a polynomial of order $2k-1$ or less.

Warped Gaussian Processes

Computation of Predictions

For any map ϕ, we can calculate explicitly

- Median: $Q_{\frac{1}{2}}(y_t) = \phi^{-1}\left(Q_{\frac{1}{2}}(x_t)\right) = \phi^{-1}(m(t))$
- Confidence intervals:
  $I^p_{y_t} = \left[\phi^{-1}(m(t) - z_p\sigma(t)), \phi^{-1}(m(t) + z_p\sigma(t))\right]$ [2]
- Sampling: $x(\mathbf{t}) \sim \mathcal{N}(m(\mathbf{t}), k(\mathbf{t}, \mathbf{t}))$ and then $y(\mathbf{t}) = \phi^{-1}(x(\mathbf{t}))$

The moments can be efficiently computed numerically using the Gauss-Hermite (**GH**) quadrature. The k-approx.[3] of the mean of y_t is

$$\mathbb{E}[y_t] = \int \phi^{-1}(x)\, p_{x_t}(x)\, dx \approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^{k} w_i \phi^{-1}\left(\sqrt{2}\sigma(t)x_i + m(t)\right)$$

where $m(t)$ and $\sigma(t)$ are the mean and std. dev. of the latent x_t, and the weights $\{w_i\}_{i=1}^k$ and locations $\{x_i\}_{i=1}^k$ are given by **GH** quadrature.

[2] $\sigma(t) = \sqrt{k(t,t)}$ and z_p is the p-quantile of standard normal (ex. $z_{0.975} \approx 1.96$)

[3] It is exact when the integrand is a polynomial of order $2k-1$ or less.

Warped Gaussian Processes

Computation of Predictions

For any map ϕ, we can calculate explicitly

- Median: $Q_{\frac{1}{2}}(y_t) = \phi^{-1}\left(Q_{\frac{1}{2}}(x_t)\right) = \phi^{-1}(m(t))$
- Confidence intervals:
  $I^p_{y_t} = \left[\phi^{-1}(m(t) - z_p\sigma(t)), \phi^{-1}(m(t) + z_p\sigma(t))\right]$ [2]
- Sampling: $x(\mathbf{t}) \sim \mathcal{N}(m(\mathbf{t}), k(\mathbf{t}, \mathbf{t}))$ and then $y(\mathbf{t}) = \phi^{-1}(x(\mathbf{t}))$

The moments can be efficiently computed numerically using the Gauss-Hermite (**GH**) quadrature. The k-approx.[3] of the mean of y_t is

$$\mathbb{E}[y_t] = \int \phi^{-1}(x)\, p_{x_t}(x)\, dx \approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^{k} w_i \phi^{-1}\left(\sqrt{2}\sigma(t)x_i + m(t)\right)$$

where $m(t)$ and $\sigma(t)$ are the mean and std. dev. of the latent x_t, and the weights $\{w_i\}_{i=1}^k$ and locations $\{x_i\}_{i=1}^k$ are given by **GH** quadrature.

---

[2] $\sigma(t) = \sqrt{k(t,t)}$ and z_p is the p-quantile of standard normal (ex. $z_{0.975} \approx 1.96$)

[3] It is exact when the integrand is a polynomial of order $2k-1$ or less.

Warped Gaussian Processes

Computation of Predictions

For any map ϕ, we can calculate explicitly

- Median: $Q_{\frac{1}{2}}(y_t) = \phi^{-1}\left(Q_{\frac{1}{2}}(x_t)\right) = \phi^{-1}(m(t))$
- Confidence intervals:
  $I_{y_t}^p = \left[\phi^{-1}(m(t) - z_p\sigma(t)), \phi^{-1}(m(t) + z_p\sigma(t))\right]$ [2]
- Sampling: $x(\mathbf{t}) \sim \mathcal{N}(m(\mathbf{t}), k(\mathbf{t}, \mathbf{t}))$ and then $y(\mathbf{t}) = \phi^{-1}(x(\mathbf{t}))$

The moments can be efficiently computed numerically using the Gauss-Hermite (**GH**) quadrature. The k-approx.[3] of the mean of y_t is

$$\mathbb{E}[y_t] = \int \phi^{-1}(x)\, p_{x_t}(x)\, dx \approx \frac{1}{\sqrt{\pi}} \sum_{i=1}^{k} w_i \phi^{-1}\left(\sqrt{2}\sigma(t) x_i + m(t)\right)$$

where $m(t)$ and $\sigma(t)$ are the mean and std. dev. of the latent x_t, and the weights $\{w_i\}_{i=1}^k$ and locations $\{x_i\}_{i=1}^k$ are given by **GH** quadrature.

---

[2] $\sigma(t) = \sqrt{k(t,t)}$ and z_p is the p-quantile of standard normal (ex. $z_{0.975} \approx 1.96$)

[3] It is exact when the integrand is a polynomial of order $2k - 1$ or less.

Box-Cox Gaussian Processes

The Box-Cox transformation: The Generalized Logarithm

- The Box-Cox function is a single-parameter $\lambda \in \mathbb{R}_0^+$ mapping

Transformation	$\varphi(y)$	$\frac{d\varphi(y)}{dy}$	$\varphi^{-1}(x)$
Affine	$a + by$	b	$\frac{x-a}{b}$
Logarithm	$\log(y)$	y^{-1}	$\exp(x)$
Box-Cox	$\frac{sgn(y)\lvert y\rvert^{\lambda}-1}{\lambda}$	$\lvert y\rvert^{\lambda-1}$	$sgn(\lambda x + 1)\,\lvert\lambda x + 1\rvert^{\frac{1}{\lambda}}$

- The Box-Cox mapping φ_λ is a power transformation (good **GH**)
- $\varphi_1(y) = y - 1$ (affine) and $\lim_{\lambda \to 0} \varphi_\lambda(y) = \log(y)$ (logarithm)
- The Box-Cox Gaussian process (**BCGP**) can model a standard **GP**, a **LogGP** and everything in between!
- The mode of the induced distribution is

$$\text{mode}_{y_t} = \left[\frac{1}{2}\left(1 + \lambda m(t) + \sqrt{(1 + \lambda m(t))^2 + 4\sigma(t)^2\lambda(\lambda - 1)}\right)\right]^{\frac{1}{\lambda}}$$

Box-Cox Gaussian Processes

The Box-Cox transformation: The Generalized Logarithm

- The Box-Cox function is a single-parameter $\lambda \in \mathbb{R}_0^+$ mapping

Transformation	$\varphi(y)$	$\frac{d\varphi(y)}{dy}$	$\varphi^{-1}(x)$
Affine	$a + by$	b	$\frac{x-a}{b}$
Logarithm	$\log(y)$	y^{-1}	$\exp(x)$
Box-Cox	$\frac{sgn(y)\lvert y\rvert^{\lambda}-1}{\lambda}$	$\lvert y\rvert^{\lambda-1}$	$sgn(\lambda x + 1)\lvert\lambda x + 1\rvert^{\frac{1}{\lambda}}$

- The Box-Cox mapping φ_λ is a power transformation (good **GH**)
- $\varphi_1(y) = y - 1$ (affine) and $\lim_{\lambda \to 0} \varphi_\lambda(y) = \log(y)$ (logarithm)
- The Box-Cox Gaussian process (**BCGP**) can model a standard **GP**, a **LogGP** and everything in between!
- The mode of the induced distribution is

$$\text{mode}_{y_t} = \left[\frac{1}{2}\left(1 + \lambda m(t) + \sqrt{(1 + \lambda m(t))^2 + 4\sigma(t)^2\lambda(\lambda - 1)}\right)\right]^{\frac{1}{\lambda}}$$

Box-Cox Gaussian Processes

The Box-Cox transformation: The Generalized Logarithm

- The Box-Cox function is a single-parameter $\lambda \in \mathbb{R}_0^+$ mapping

Transformation	$\varphi(y)$	$\frac{d\varphi(y)}{dy}$	$\varphi^{-1}(x)$
Affine	$a + by$	b	$\frac{x-a}{b}$
Logarithm	$\log(y)$	y^{-1}	$\exp(x)$
Box-Cox	$\frac{sgn(y)\lvert y\rvert^{\lambda}-1}{\lambda}$	$\lvert y\rvert^{\lambda-1}$	$sgn\left(\lambda x+1\right)\lvert\lambda x+1\rvert^{\frac{1}{\lambda}}$

- The Box-Cox mapping φ_λ is a power transformation (good **GH**)
- $\varphi_1(y) = y - 1$ (affine) and $\lim_{\lambda \to 0} \varphi_\lambda(y) = \log(y)$ (logarithm)
- The Box-Cox Gaussian process (**BCGP**) can model a standard **GP**, a **LogGP** and everything in between!
- The mode of the induced distribution is

$$\text{mode}_{y_t} = \left[\frac{1}{2}\left(1 + \lambda m(t) + \sqrt{\left(1+\lambda m(t)\right)^2 + 4\sigma(t)^2\lambda\left(\lambda-1\right)}\right)\right]^{\frac{1}{\lambda}}$$

Box-Cox Gaussian Processes

The Box-Cox transformation: The Generalized Logarithm

- The Box-Cox function is a single-parameter $\lambda \in \mathbb{R}_0^+$ mapping

Transformation	$\varphi(y)$	$\frac{d\varphi(y)}{dy}$	$\varphi^{-1}(x)$
Affine	$a + by$	b	$\frac{x-a}{b}$
Logarithm	$\log(y)$	y^{-1}	$\exp(x)$
Box-Cox	$\frac{sgn(y)\lvert y\rvert^{\lambda}-1}{\lambda}$	$\lvert y\rvert^{\lambda-1}$	$sgn\left(\lambda x+1\right)\lvert\lambda x+1\rvert^{\frac{1}{\lambda}}$

- The Box-Cox mapping φ_λ is a power transformation (good **GH**)
- $\varphi_1(y) = y - 1$ (affine) and $\lim_{\lambda \to 0} \varphi_\lambda(y) = \log(y)$ (logarithm)
- The Box-Cox Gaussian process (**BCGP**) can model a standard **GP**, a **LogGP** and everything in between!
- The mode of the induced distribution is

$$\mathrm{mode}_{y_t} = \left[\frac{1}{2}\left(1 + \lambda m(t) + \sqrt{(1+\lambda m(t))^2 + 4\sigma(t)^2\lambda\left(\lambda - 1\right)}\right)\right]^{\frac{1}{\lambda}}$$

Box-Cox Gaussian Processes

The Box-Cox transformation: The Generalized Logarithm

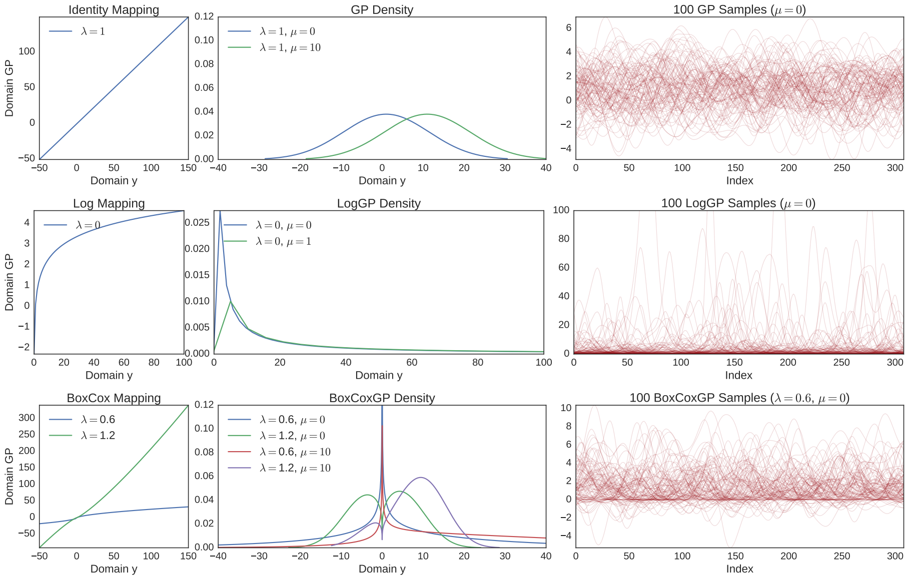
- The Box-Cox function is a single-parameter $\lambda \in \mathbb{R}_0^+$ mapping

Transformation	$\varphi(y)$	$\frac{d\varphi(y)}{dy}$	$\varphi^{-1}(x)$
Affine	$a + by$	b	$\frac{x-a}{b}$
Logarithm	$\log(y)$	y^{-1}	$\exp(x)$
Box-Cox	$\frac{sgn(y)\lvert y\rvert^{\lambda}-1}{\lambda}$	$\lvert y\rvert^{\lambda-1}$	$sgn\left(\lambda x+1\right)\lvert\lambda x+1\rvert^{\frac{1}{\lambda}}$

- The Box-Cox mapping φ_λ is a power transformation (good **GH**)
- $\varphi_1(y) = y - 1$ (affine) and $\lim_{\lambda\to 0} \varphi_\lambda(y) = \log(y)$ (logarithm)
- The Box-Cox Gaussian process (**BCGP**) can model a standard **GP**, a **LogGP** and everything in between!
- The mode of the induced distribution is
$$\text{mode}_{y_t} = \left[\frac{1}{2}\left(1 + \lambda m(t) + \sqrt{(1+\lambda m(t))^2 + 4\sigma(t)^2\lambda\left(\lambda-1\right)}\right)\right]^{\frac{1}{\lambda}}$$

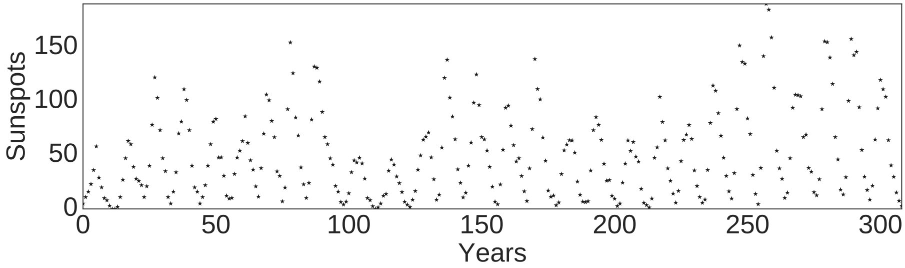
Box-Cox Gaussian Processes

A Flexible and Tractable Non-Gaussian Process



Box-Cox Gaussian Processes

Reconstruction and forecasting of the Sunspots time series

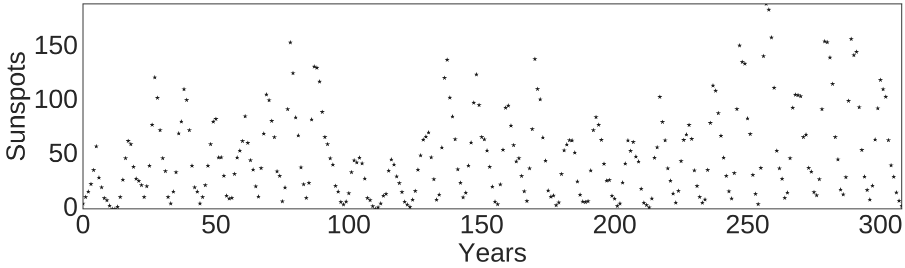


Sunspot time series between 1700 and 2008 (309 points)

- Positive almost-periodic time series
- Training with 131 random observations before 1961
- Standard **GP** vs **Box-Cox GP** with 2-component SM kernel
- **BFGS** vs **Hybrid BFGS-Powell** for training hyperparameters
- Reconstructing the signal before 1961 (131 datapoints)
- Forecasting the signal after 1961 (47 datapoints)
- Performance evaluated with **MAE**, **MSE** and **NLPD** scores

Box-Cox Gaussian Processes

Reconstruction and forecasting of the Sunspots time series

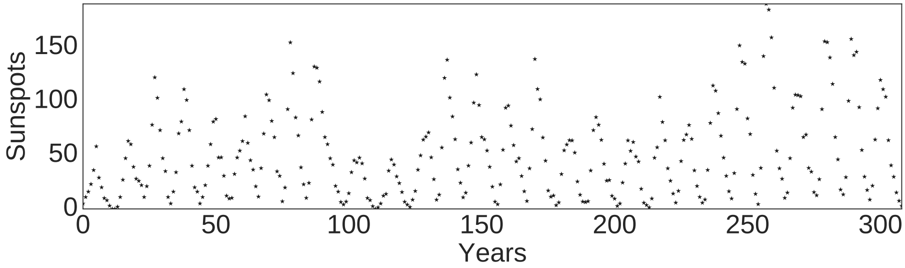


Sunspot time series between 1700 and 2008 (309 points)

- Positive almost-periodic time series
- Training with 131 random observations before 1961
- Standard **GP** vs **Box-Cox GP** with 2-component SM kernel
- **BFGS** vs **Hybrid BFGS-Powell** for training hyperparameters
- Reconstructing the signal before 1961 (131 datapoints)
- Forecasting the signal after 1961 (47 datapoints)
- Performance evaluated with **MAE**, **MSE** and **NLPD** scores

Box-Cox Gaussian Processes

Reconstruction and forecasting of the Sunspots time series

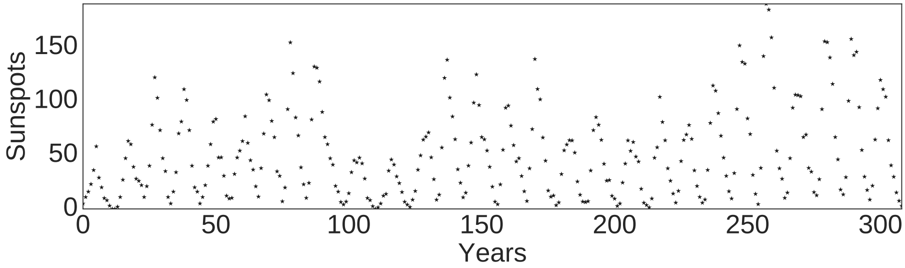


Sunspot time series between 1700 and 2008 (309 points)

- Positive almost-periodic time series
- Training with 131 random observations before 1961
- Standard **GP** vs **Box-Cox GP** with 2-component SM kernel
- **BFGS** vs **Hybrid BFGS-Powell** for training hyperparameters
- Reconstructing the signal before 1961 (131 datapoints)
- Forecasting the signal after 1961 (47 datapoints)
- Performance evaluated with **MAE**, **MSE** and **NLPD** scores

Box-Cox Gaussian Processes

Reconstruction and forecasting of the Sunspots time series

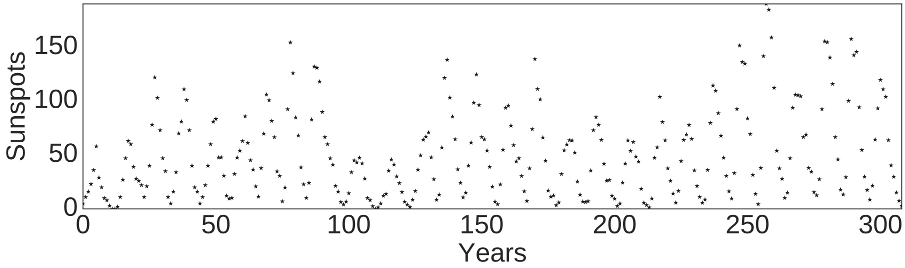


Sunspot time series between 1700 and 2008 (309 points)

- Positive almost-periodic time series
- Training with 131 random observations before 1961
- Standard **GP** vs **Box-Cox GP** with 2-component SM kernel
- **BFGS** vs **Hybrid BFGS-Powell** for training hyperparameters
- Reconstructing the signal before 1961 (131 datapoints)
- Forecasting the signal after 1961 (47 datapoints)
- Performance evaluated with **MAE**, **MSE** and **NLPD** scores

Box-Cox Gaussian Processes

Reconstruction and forecasting of the Sunspots time series

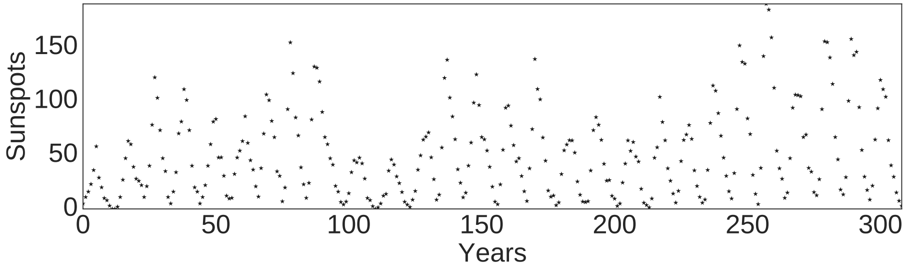


Sunspot time series between 1700 and 2008 (309 points)

- Positive almost-periodic time series
- Training with 131 random observations before 1961
- Standard **GP** vs **Box-Cox GP** with 2-component SM kernel
- **BFGS** vs **Hybrid BFGS-Powell** for training hyperparameters
- Reconstructing the signal before 1961 (131 datapoints)
- Forecasting the signal after 1961 (47 datapoints)
- Performance evaluated with **MAE**, **MSE** and **NLPD** scores

Box-Cox Gaussian Processes

Reconstruction and forecasting of the Sunspots time series

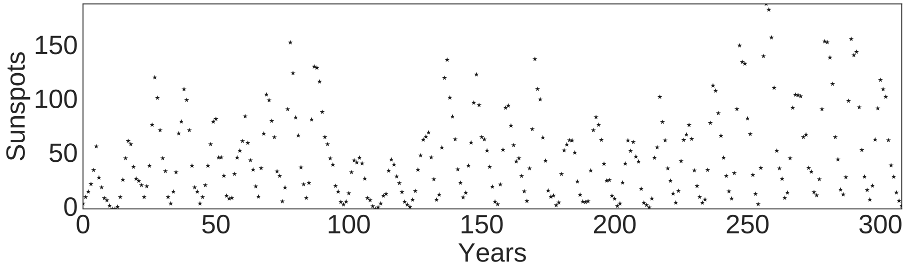


Sunspot time series between 1700 and 2008 (309 points)

- Positive almost-periodic time series
- Training with 131 random observations before 1961
- Standard **GP** vs **Box-Cox GP** with 2-component SM kernel
- **BFGS** vs **Hybrid BFGS-Powell** for training hyperparameters
- Reconstructing the signal before 1961 (131 datapoints)
- Forecasting the signal after 1961 (47 datapoints)
- Performance evaluated with **MAE**, **MSE** and **NLPD** scores

Box-Cox Gaussian Processes

Reconstruction and forecasting of the Sunspots time series

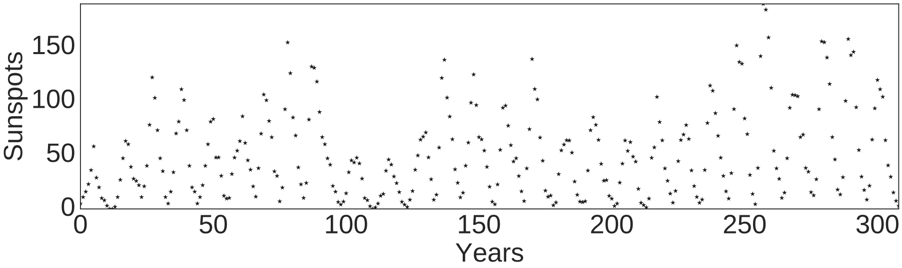


Sunspot time series between 1700 and 2008 (309 points)

- Positive almost-periodic time series
- Training with 131 random observations before 1961
- Standard **GP** vs **Box-Cox GP** with 2-component SM kernel
- **BFGS** vs **Hybrid BFGS-Powell** for training hyperparameters
- Reconstructing the signal before 1961 (131 datapoints)
- Forecasting the signal after 1961 (47 datapoints)
- Performance evaluated with **MAE**, **MSE** and **NLPD** scores

Box-Cox Gaussian Processes

Reconstruction and forecasting of the Sunspots time series

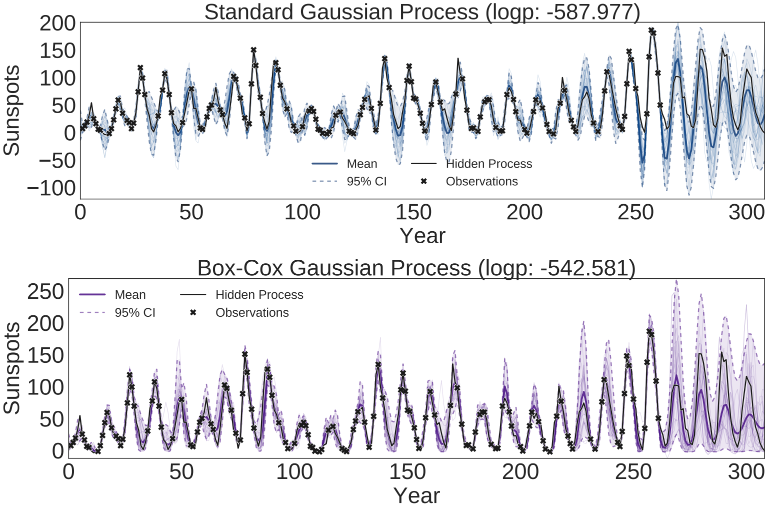


Sunspot time series between 1700 and 2008 (309 points)

- Positive almost-periodic time series
- Training with 131 random observations before 1961
- Standard **GP** vs **Box-Cox GP** with 2-component SM kernel
- **BFGS** vs **Hybrid BFGS-Powell** for training hyperparameters
- Reconstructing the signal before 1961 (131 datapoints)
- Forecasting the signal after 1961 (47 datapoints)
- Performance evaluated with **MAE**, **MSE** and **NLPD** scores

Box-Cox Gaussian Processes

Reconstruction and forecasting of the Sunspots time series



Reconstruction and forecasting of the Sunspot series using GP (top) and BCGP (bottom) trained using BFGS-Powell.

Box-Cox Gaussian Processes

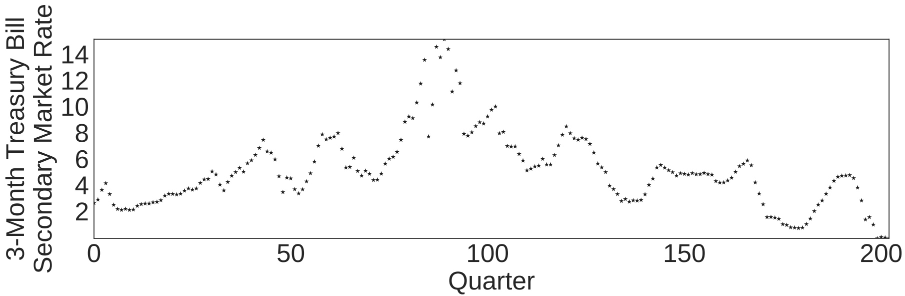
Reconstruction and forecasting of the Sunspots time series

		MAE	MSE	NLPD	NLL
Reconst.	GP BFGS	11.06	237.19	4.06	608.27
	GP BFGS-Powell	10.37	217.96	4.03	587.98
	BCGP BFGS	11.06	239.36	4.03	578.68
	BCGP BFGS-Powell	**8.85**	**150.36**	**3.90**	**542.58**
Forecast	GP BFGS	40.36	2509.55	5.36	608.27
	GP BFGS-Powell	30.68	1414.81	5.17	587.98
	BCGP BFGS	40.25	2526.24	5.20	578.68
	BCGP BFGS-Powell	**26.90**	**1253.10**	**4.95**	**542.58**

Performance of GP and BCGP for reconstruction and forecasting of the Sunspots data trained using BFGS and BFGS-Powell.

Box-Cox Gaussian Processes

Learning Macroeconomic time series

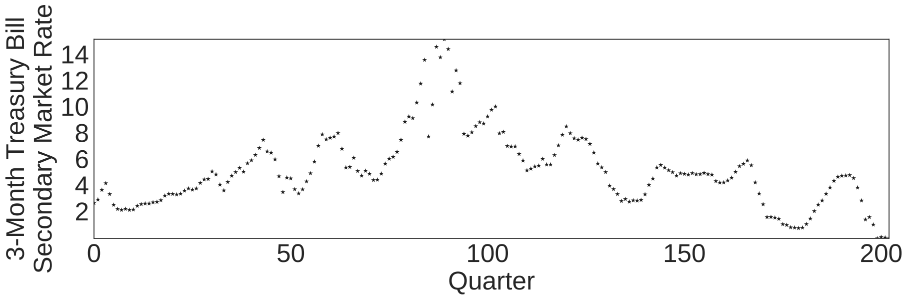


Quarterly average *3-Month Treasury Bill: Secondary Market Rate* between 1959 and 2009, representing the price of U.S. government risk-free bonds.

- Non-negative values and large positive deviations
- Training with 30 datapoints (15%)
- Standard **GP** vs **Box-Cox GP** with square exponential kernel
- **Hybrid BFGS-Powell** vs **MCMC** for training hyperparameters

Box-Cox Gaussian Processes

Learning Macroeconomic time series

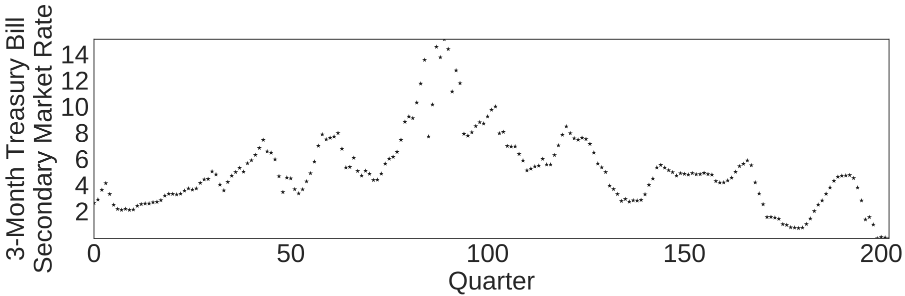


Quarterly average *3-Month Treasury Bill: Secondary Market Rate* between 1959 and 2009, representing the price of U.S. government risk-free bonds.

- Non-negative values and large positive deviations
- Training with 30 datapoints (15%)
- Standard **GP** vs **Box-Cox GP** with square exponential kernel
- **Hybrid BFGS-Powell** vs **MCMC** for training hyperparameters

Box-Cox Gaussian Processes

Learning Macroeconomic time series

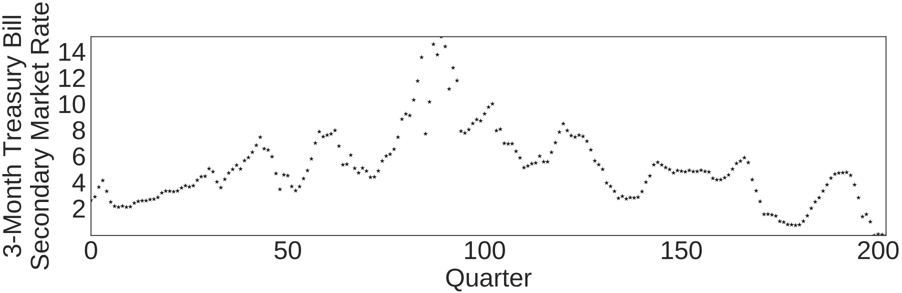


Quarterly average *3-Month Treasury Bill: Secondary Market Rate* between 1959 and 2009, representing the price of U.S. government risk-free bonds.

- Non-negative values and large positive deviations
- Training with 30 datapoints (15%)
- Standard **GP** vs **Box-Cox GP** with square exponential kernel
- **Hybrid BFGS-Powell** vs **MCMC** for training hyperparameters

Box-Cox Gaussian Processes

Learning Macroeconomic time series

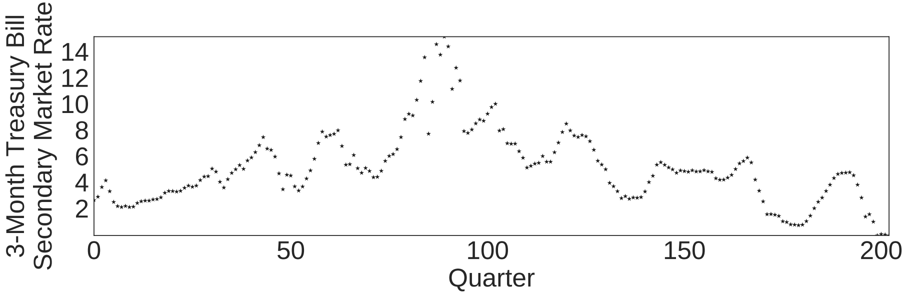


Quarterly average *3-Month Treasury Bill: Secondary Market Rate* between 1959 and 2009, representing the price of U.S. government risk-free bonds.

- Non-negative values and large positive deviations
- Training with 30 datapoints (15%)
- Standard **GP** vs **Box-Cox GP** with square exponential kernel
- **Hybrid BFGS-Powell** vs **MCMC** for training hyperparameters

Box-Cox Gaussian Processes

Learning Macroeconomic time series

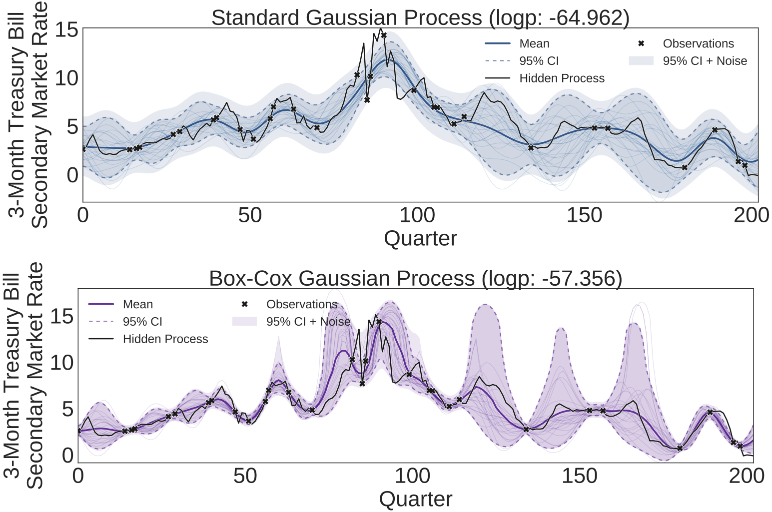


Quarterly average *3-Month Treasury Bill: Secondary Market Rate* between 1959 and 2009, representing the price of U.S. government risk-free bonds.

- Non-negative values and large positive deviations
- Training with 30 datapoints (15%)
- Standard **GP** vs **Box-Cox GP** with square exponential kernel
- **Hybrid BFGS-Powell** vs **MCMC** for training hyperparameters

Box-Cox Gaussian Processes

Learning Macroeconomic time series



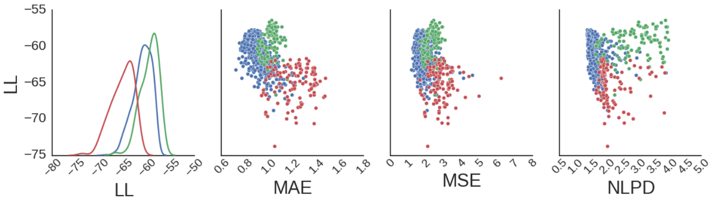
Standard GP (top) and Box-Cox GP (bottom) trained using the ensemble MCMC method on a macroeconomic time series.

Box-Cox Gaussian Processes

Learning Macroeconomic time series

	MAE	MSE	NLPD	NLL
GP BFGS-Powell	1.28	2.83	1.94	64.27
GP MCMC	0.95	1.79	1.74	64.96
BCGP BFGS-Powell	0.93	1.94	1.69	59.21
BCGP MCMC	**0.88**	**1.75**	**1.42**	**57.36**

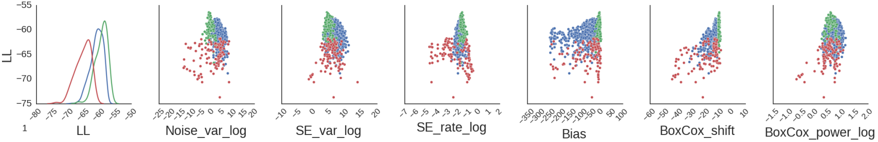
Performance of GP and BCGP for reconstruction of macroeconomic data trained using BFGS-Powell and MCMC.



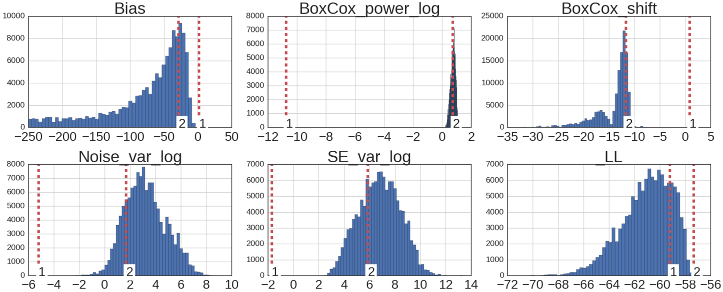
Log-likelihood against scores for BCGP on macroeconomic data.

Box-Cox Gaussian Processes

Learning Macroeconomic time series



Scatter plot of BCGP hyperparameters against their log-likelihood.



Marginal densities of hyperparameters in cluster blue.

Line 1: BFGS-Powell model – Line 2: MCMC model.

Box-Cox Gaussian Processes

Discussion

- **GP** is a generative model for time series with closed-form formulas for training and prediction.
- Real-world time series not necessarily normally distributed
- **Warped GP** is a formal recipe to construct non-Gaussian models
- **Box-Cox GP** has the ability to discover non-Gaussian features
- Gradient-based method **BFGS** has lower performance on training than the derivative-free methods as **Powell** and **MCMC**
- Further research towards more expressive transformations
  - Gonzalo Rios and Felipe Tobar. Learning non-Gaussian Time Series using the Box-Cox Gaussian Process. arXiv preprint arxiv.org/abs/1803.07102 (2018).
  - Gonzalo Rios and Felipe Tobar. Compositionally-Warped Gaussian Processes. Under review at IEEE Transactions on Neural Networks and Learning System (2018).

Box-Cox Gaussian Processes

Discussion

- **GP** is a generative model for time series with closed-form formulas for training and prediction.
- Real-world time series not necessarily normally distributed
- **Warped GP** is a formal recipe to construct non-Gaussian models
- **Box-Cox GP** has the ability to discover non-Gaussian features
- Gradient-based method **BFGS** has lower performance on training than the derivative-free methods as **Powell** and **MCMC**
- Further research towards more expressive transformations

  - Gonzalo Rios and Felipe Tobar. Learning non-Gaussian Time Series using the Box-Cox Gaussian Process. arXiv preprint arxiv.org/abs/1803.07102 (2018).
  - Gonzalo Rios and Felipe Tobar. Compositionally-Warped Gaussian Processes. Under review at IEEE Transactions on Neural Networks and Learning System (2018).

Box-Cox Gaussian Processes

Discussion

- **GP** is a generative model for time series with closed-form formulas for training and prediction.
- Real-world time series not necessarily normally distributed
- **Warped GP** is a formal recipe to construct non-Gaussian models
- **Box-Cox GP** has the ability to discover non-Gaussian features
- Gradient-based method **BFGS** has lower performance on training than the derivative-free methods as **Powell** and **MCMC**
- Further research towards more expressive transformations

  - Gonzalo Rios and Felipe Tobar. Learning non-Gaussian Time Series using the Box-Cox Gaussian Process. arXiv preprint arxiv.org/abs/1803.07102 (2018).
  - Gonzalo Rios and Felipe Tobar. Compositionally-Warped Gaussian Processes. Under review at IEEE Transactions on Neural Networks and Learning System (2018).

Box-Cox Gaussian Processes

Discussion

- **GP** is a generative model for time series with closed-form formulas for training and prediction.
- Real-world time series not necessarily normally distributed
- **Warped GP** is a formal recipe to construct non-Gaussian models
- **Box-Cox GP** has the ability to discover non-Gaussian features
- Gradient-based method **BFGS** has lower performance on training than the derivative-free methods as **Powell** and **MCMC**
- Further research towards more expressive transformations
  - Gonzalo Rios and Felipe Tobar. Learning non-Gaussian Time Series using the Box-Cox Gaussian Process. arXiv preprint arxiv.org/abs/1803.07102 (2018).
  - Gonzalo Rios and Felipe Tobar. Compositionally-Warped Gaussian Processes. Under review at IEEE Transactions on Neural Networks and Learning System (2018).

Box-Cox Gaussian Processes

Discussion

- **GP** is a generative model for time series with closed-form formulas for training and prediction.
- Real-world time series not necessarily normally distributed
- **Warped GP** is a formal recipe to construct non-Gaussian models
- **Box-Cox GP** has the ability to discover non-Gaussian features
- Gradient-based method **BFGS** has lower performance on training than the derivative-free methods as **Powell** and **MCMC**
- Further research towards more expressive transformations

  - Gonzalo Rios and Felipe Tobar. Learning non-Gaussian Time Series using the Box-Cox Gaussian Process. arXiv preprint arxiv.org/abs/1803.07102 (2018).
  - Gonzalo Rios and Felipe Tobar. Compositionally-Warped Gaussian Processes. Under review at IEEE Transactions on Neural Networks and Learning System (2018).

Box-Cox Gaussian Processes

Discussion

- **GP** is a generative model for time series with closed-form formulas for training and prediction.
- Real-world time series not necessarily normally distributed
- **Warped GP** is a formal recipe to construct non-Gaussian models
- **Box-Cox GP** has the ability to discover non-Gaussian features
- Gradient-based method **BFGS** has lower performance on training than the derivative-free methods as **Powell** and **MCMC**
- Further research towards more expressive transformations

  - Gonzalo Rios and Felipe Tobar. Learning non-Gaussian Time Series using the Box-Cox Gaussian Process. arXiv preprint arxiv.org/abs/1803.07102 (2018).
  - Gonzalo Rios and Felipe Tobar. Compositionally-Warped Gaussian Processes. Under review at IEEE Transactions on Neural Networks and Learning System (2018).

Box-Cox Gaussian Processes

Discussion

- **GP** is a generative model for time series with closed-form formulas for training and prediction.
- Real-world time series not necessarily normally distributed
- **Warped GP** is a formal recipe to construct non-Gaussian models
- **Box-Cox GP** has the ability to discover non-Gaussian features
- Gradient-based method **BFGS** has lower performance on training than the derivative-free methods as **Powell** and **MCMC**
- Further research towards more expressive transformations
  - Gonzalo Rios and Felipe Tobar. Learning non-Gaussian Time Series using the Box-Cox Gaussian Process. arXiv preprint arxiv.org/abs/1803.07102 (2018).
  - Gonzalo Rios and Felipe Tobar. Compositionally-Warped Gaussian Processes. Under review at IEEE Transactions on Neural Networks and Learning System (2018).

Box-Cox Gaussian Processes

Discussion

- **GP** is a generative model for time series with closed-form formulas for training and prediction.
- Real-world time series not necessarily normally distributed
- **Warped GP** is a formal recipe to construct non-Gaussian models
- **Box-Cox GP** has the ability to discover non-Gaussian features
- Gradient-based method **BFGS** has lower performance on training than the derivative-free methods as **Powell** and **MCMC**
- Further research towards more expressive transformations
  - Gonzalo Rios and Felipe Tobar. Learning non-Gaussian Time Series using the Box-Cox Gaussian Process. arXiv preprint arxiv.org/abs/1803.07102 (2018).
  - Gonzalo Rios and Felipe Tobar. Compositionally-Warped Gaussian Processes. Under review at IEEE Transactions on Neural Networks and Learning System (2018).

Box-Cox Gaussian Processes

Discussion

- **GP** is a generative model for time series with closed-form formulas for training and prediction.
- Real-world time series not necessarily normally distributed
- **Warped GP** is a formal recipe to construct non-Gaussian models
- **Box-Cox GP** has the ability to discover non-Gaussian features
- Gradient-based method **BFGS** has lower performance on training than the derivative-free methods as **Powell** and **MCMC**
- Further research towards more expressive transformations

  - Gonzalo Rios and Felipe Tobar. Learning non-Gaussian Time Series using the Box-Cox Gaussian Process. arXiv preprint arxiv.org/abs/1803.07102 (2018).
  - Gonzalo Rios and Felipe Tobar. Compositionally-Warped Gaussian Processes. Under review at IEEE Transactions on Neural Networks and Learning System (2018).

Thanks!

Questions?