Gaussian Processes Part 2

Max Kochurov

Moscow State University

Lecture 5

Agenda

Time Series, Classical Approach

If data has seasonality, you usually use STL decomposition. However,

- Parameters are not interpretable, only decomposition is available
- No uncertainty estimates
- Quite strict on input values
- Significantly less flexible in modelling

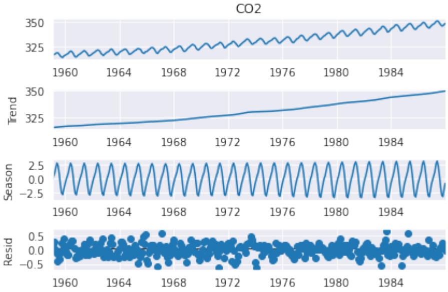


Figure: STL decomposition for CO2 data, Statsmodels

GP decomposition

A Gaussian process can handle a complicated set of assumptions in addition to what STL provides

- Granular seasonality (year + quarter + month + week)
- Changepoint models
- Flexible likelihood Function
- Panel regression models
- Missing values

Typical Model

Typical model is additive

$$x_t \sim \underbrace{g(t)}_{non-periodic} + \underbrace{s(t)}_{periodic} + \underbrace{h(t)}_{holidays}$$

Reference

See more in Prophet preprint [2]. Every time series model is unique

Reminder

$x \in \mathbb{R}^n$, $y \in \mathbb{R}$

$$Y \sim \mathcal{GP}(m(x), k(x, x'))$$

Reminder

$x \in \mathbb{R}^n$, $y \in \mathbb{R}$

$$Y \sim \mathcal{GP}(m(x), k(x, x'))$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_N) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{bmatrix} \right)$$

1. $\mathcal{GP}$ Gaussian Process - simply, a normal distribution with special mean $m(x)$ and covariance $k(x, x')$

Reminder

$x \in \mathbb{R}^n$, $y \in \mathbb{R}$

$$Y \sim \mathcal{GP}(m(x), k(x, x'))$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_N) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{bmatrix}\right)$$

1. $\mathcal{GP}$ Gaussian Process - simply, a normal distribution with special mean $m(x)$ and covariance $k(x, x')$
2. $m(x)$ - mean function, e.g.
   - Linear regression $m(x) = x^\top \beta$
   - Simply Constant or Zero $m(x) = c$
   - Other custom functions $m(x) = \sin(x)$

Reminder

$x \in \mathbb{R}^n$, $y \in \mathbb{R}$

$$Y \sim \mathcal{GP}(m(x), k(x, x'))$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_N) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{bmatrix} \right)$$

1. $\mathcal{GP}$ Gaussian Process - simply, a normal distribution with special mean $m(x)$ and covariance $k(x, x')$
2. $m(x)$ - mean function, e.g.
   - Linear regression $m(x) = x^\top \beta$
   - Simply Constant or Zero $m(x) = c$
   - Other custom functions $m(x) = \sin(x)$
3. $k(x, x')$ - kernel function, simply - measure of similarity for x and x'
   - $[K]_{ij} = k(x_i, x_j)$ is an SPD matrix

Non-periodic Part (mean function)

- Growth models
- Linear trend models
- Changepoint models

Non-periodic Part (mean function)

- Growth models
- Linear trend models
- Changepoint models

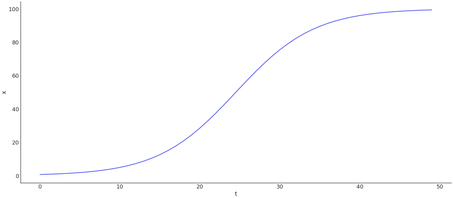


Figure: Growth Model

$$x = \frac{c}{1 + \exp(-k(t - m))}$$

Non-periodic Part (mean function)

- Growth models
- Linear trend models
- Changepoint models

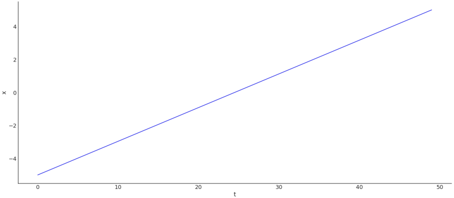


Figure: Linear Trend Model

$$x = \frac{c}{1 + \exp(-k(t - m))}$$

Non-periodic Part (mean function)

- Growth models
- Linear trend models
- Changepoint models

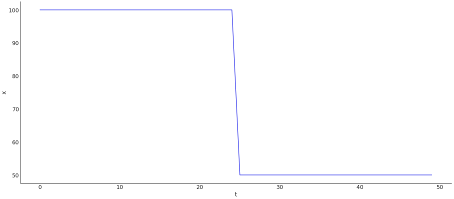


Figure: Changepoint Model

$$x = \begin{cases} c_1, & t < m \\ c_2, & t \geq m \end{cases}$$

Non-periodic Part (mean function)

- Growth models
- Linear trend models
- Changepoint models

Extentions

Extensions are possible, e.g. time dependent saturation in the growth model. See in [2]

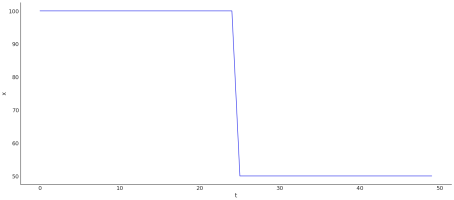


Figure: Changepoint Model

$$x = \begin{cases} c_1, & t < m \\ c_2, & t \geq m \end{cases}$$

Holidays

$$h(t) = \text{is-holiday}(t)$$

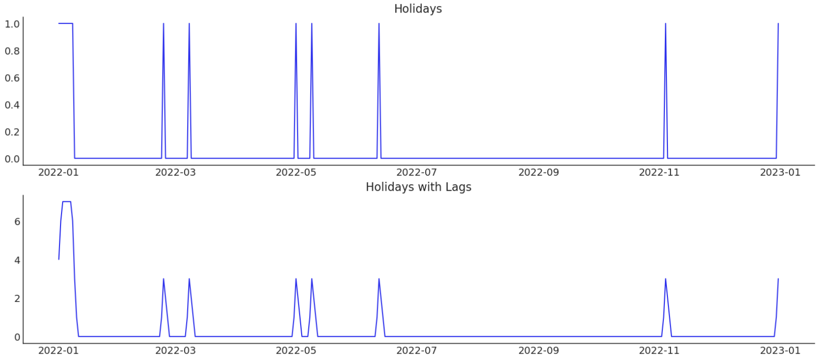


Figure: Holiday features

Periodic part (cov function)

Granularities are important here. Multiple Periodic kernels can be used.

- Yearly
- Quarterly
- Monthly
- Weekly

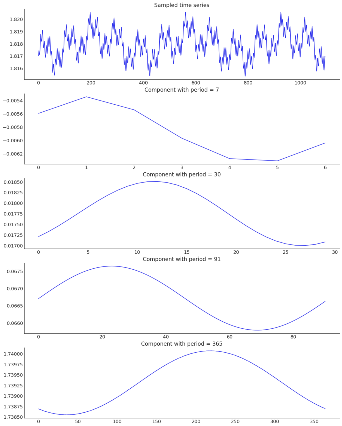


Figure: Seasonal Decomposition

Lengthscales for Periodic Part

Hyperparameters

Common sense driven lenthscale choice

- Week - couple of days make a change ($ls \approx 3$)
- Month - week makes sense for a dramatic change ($ls \approx 7$)
- Quarter - month makes sense for a dramatic change ($ls \approx 30$)
- Year - quarter makes sense for a dramatic change ($ls \approx 90$)

Lengthscales for Periodic Part

Hyperparameters

Common sense driven lenthscale choice

- Week - couple of days make a change ($ls \approx 3$)
- Month - week makes sense for a dramatic change ($ls \approx 7$)
- Quarter - month makes sense for a dramatic change ($ls \approx 30$)
- Year - quarter makes sense for a dramatic change ($ls \approx 90$)

In practice

Everybody is using Fourier features as a replacement for Periodic Kernel

Putting All Together

$$m(t) = \underbrace{g(t)}_{non-periodic} + \underbrace{h(t)}_{holidays}$$

$$\begin{aligned} k(*,*) = & \ \mathrm{Periodic}(\mathrm{p} = 365, \mathrm{l} = 90) \\ + & \ \mathrm{Periodic}(\mathrm{p} = 90, \mathrm{l} = 30) \\ + & \ \mathrm{Periodic}(\mathrm{p} = 30, \mathrm{l} = 7) \\ + & \ \mathrm{Periodic}(\mathrm{p} = 7, \mathrm{l} = 3) \end{aligned}$$

Putting All Together

$$m(t) = \underbrace{g(t)}_{non-periodic} + \underbrace{h(t)}_{holidays}$$

$$\begin{aligned} k(*,*) = & \; \alpha_{365}\,\text{Periodic}(\text{p} = 365, \text{l} = 90) \\ & + \alpha_{90}\,\text{Periodic}(\text{p} = 90, \text{l} = 30) \\ & + \alpha_{30}\,\text{Periodic}(\text{p} = 30, \text{l} = 7) \\ & + \alpha_{7}\,\text{Periodic}(\text{p} = 7, \text{l} = 3) \end{aligned}$$

Missing Parts

1. Weights for periodic components

Putting All Together

$$m(t) = \underbrace{g(t)}_{non-periodic} + \underbrace{h(t)}_{holidays}$$

$$\begin{aligned} k(*,*) = & \alpha_{365}\,\mathrm{Periodic}(\mathrm{p}=365, \mathrm{l}=90) \\ & + \alpha_{90}\,\mathrm{Periodic}(\mathrm{p}=90, \mathrm{l}=30) \\ & + \alpha_{30}\,\mathrm{Periodic}(\mathrm{p}=30, \mathrm{l}=7) \\ & + \alpha_{7}\,\mathrm{Periodic}(\mathrm{p}=7, \mathrm{l}=3) \\ & + \beta\,\mathrm{ExpQuad}(\mathrm{l}=90) + \mathrm{WhiteNoise}(\gamma) \end{aligned}$$

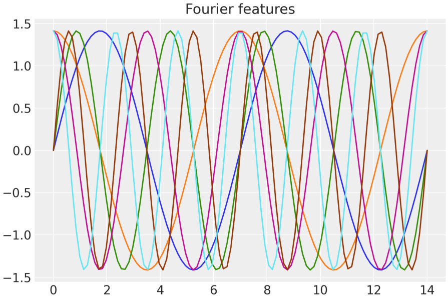
Missing Parts

1. Weights for periodic components
2. Trent violations from $g(t)$

More Efficient Periodic

Having at least one Periodic kernel in a large time series prevents from optimizations.

- Fourier features can be added as regressors
- This allows reasonable periodicity to be present in the model



Choosing Features

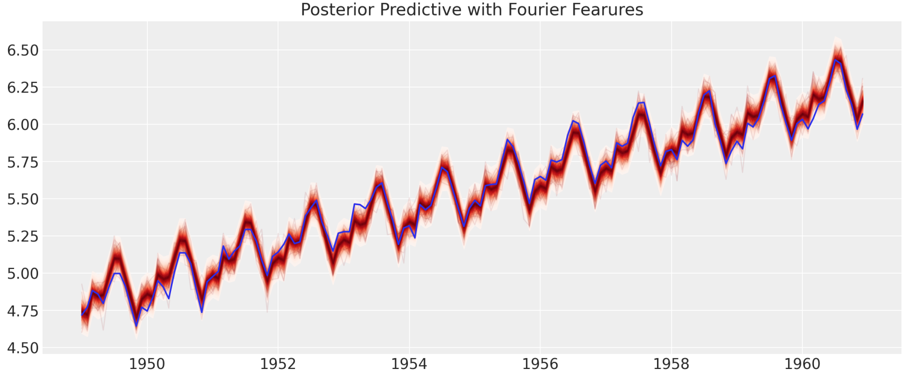
Every regular period has optimal number of fourier components (aka order)

- Weekly: 3
- Monthly: 10
- Yearly: 5

```
from collections import namedtuple
from enum import Enum

Season = namedtuple("Season", "period,order")

class Daily(Season, Enum):
    Week = 7.0, 3
    Year = 365.25, 5
    Month = 365.25 / 12, 10

class Monthly(Season, Enum):
    Year = 12, 5
    Month = 1, 10
```

Applications of Fourier Features

- E.g. Prophet uses Fourier Features (FF)
- It is VERY Fast, drop-in replacement for Periodic GP
- Proven to be useful for Bayesian models



Integrating into the model

```
alpha = pm.Normal("alpha", 0, sigma, shape=Monthly.Year.order * 2)
features = fourier_series(months, season=Monthly.Year)
seasonality = features @ alpha

expectation = bias + trend + seasonality
```

Stochastic Volatility Model

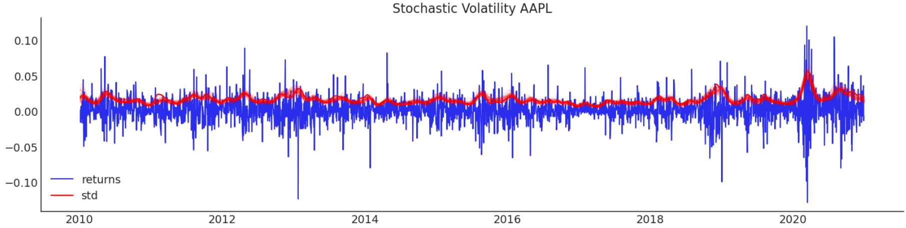


Figure: Stochastic Volatility Estimation for AAPL

Read More

https://github.com/quantopian/bayesalpha

Key Ideas

- Returns are constant
- Volatility is not constant
- Model volatility as a Gaussian process
- Use approximations to speed up the model
- Use GPU to do fast inference

Priors

- Returns

- Volatility

- Time Component

Priors

- Returns
  - Expect year return at orders $\pm 100\%$
- Volatility
- Time Component

Priors

- Returns
  - Expect year return at orders $\pm 100\%$
  - Daily return $\pm(2^{\frac{1}{250}} - 1)$
- Volatility
- Time Component

Priors

- Returns
  - Expect year return at orders $\pm 100\%$
  - Daily return $\pm(2^{\frac{1}{250}} - 1)$
  - The prior is insane but yet motivated, name your prior!
- Volatility
- Time Component

Priors

- Returns
  - Expect year return at orders $\pm 100\%$
  - Daily return $\pm(2^{\frac{1}{250}} - 1)$
  - The prior is insane but yet motivated, name your prior!
- Volatility
  - Expect signal to noise $\frac{\text{mean}}{\text{std}} \approx 1$
- Time Component

Priors

- Returns
  - Expect year return at orders $\pm 100\%$
  - Daily return $\pm(2^{\frac{1}{250}} - 1)$
  - The prior is insane but yet motivated, name your prior!
- Volatility
  - Expect signal to noise $\frac{\text{mean}}{\text{std}} \approx 1$
  - $\log \text{std} = \log(2^{\frac{1}{250}} - 1)$
- Time Component

Priors

- Returns
  - Expect year return at orders $\pm 100\%$
  - Daily return $\pm(2^{\frac{1}{250}} - 1)$
  - The prior is insane but yet motivated, name your prior!
- Volatility
  - Expect signal to noise $\frac{\text{mean}}{\text{std}} \approx 1$
  - $\log \text{std} = \log(2^{\frac{1}{250}} - 1)$
- Time Component
  - After the model is framed

The Model

I did the modelling choice to make it as simple as possible

$$
\begin{aligned}
\text{return} &\sim \text{Normal}(0, 2^{\frac{1}{250}} - 1) \\
\log \text{std} &\sim \text{Normal}(\log(2^{\frac{1}{250}} - 1), 0.05) \\
\text{ls} &\sim \text{Gamma}(30, 5) \\
\alpha_{\text{vol}} &\sim \text{Exponential}(100) \\
K(*, *) &= \alpha_{\text{vol}}\, \text{Matern32}(\text{ls}) \\
\Delta_t^{\log \text{std}} &\sim \mathcal{GP}(0, K) \\
\text{obs}_t &\sim \text{Normal}(\text{return}, \exp(\log \text{std} + \Delta_t^{\log \text{std}}))
\end{aligned}
$$

The Model

I did the modelling choice to make it as simple as possible

$$
\begin{aligned}
\text{return} &\sim \text{Normal}(0, 2^{\frac{1}{250}} - 1) \\
\log \text{std} &\sim \text{Normal}(\log(2^{\frac{1}{250}} - 1), 0.05) \\
\text{ls} &\sim \text{Gamma}(30, 5) \\
\alpha_{\text{vol}} &\sim \text{Exponential}(100) \\
K(*, *) &= \alpha_{\text{vol}} \, \text{Matern32}(\text{ls}) \\
\Delta_t^{\log \text{std}} &\sim \mathcal{GP}(0, K) \\
\text{obs}_t &\sim \text{Normal}(\text{return}, \exp(\log \text{std} + \Delta_t^{\log \text{std}}))
\end{aligned}
$$

The Model

I did the modelling choice to make it as simple as possible

$$
\begin{aligned}
\text{return} &\sim \text{Normal}(0, 2^{\frac{1}{250}} - 1) \\
\log \text{std} &\sim \text{Normal}(\log(2^{\frac{1}{250}} - 1), 0.05) \\
\text{ls} &\sim \text{Gamma}(30, 5) \\
\alpha_{\text{vol}} &\sim \text{Exponential}(100) \\
K(*,*) &= \alpha_{\text{vol}} \, \text{Matern32}(\text{ls}) \\
\Delta_t^{\log \text{std}} &\sim \mathcal{GP}(0, K) \\
\text{obs}_t &\sim \text{Normal}(\text{return}, \exp(\log \text{std} + \Delta_t^{\log \text{std}}))
\end{aligned}
$$

The Model

I did the modelling choice to make it as simple as possible

$$
\begin{aligned}
\text{return} &\sim \text{Normal}(0, 2^{\frac{1}{250}} - 1) \\
\log \text{std} &\sim \text{Normal}(\log(2^{\frac{1}{250}} - 1), 0.05) \\
\text{ls} &\sim \text{Gamma}(30, 5) \\
\alpha_{\text{vol}} &\sim \text{Exponential}(100) \\
K(*, *) &= \alpha_{\text{vol}} \,\text{Matern32}(\text{ls}) \\
\Delta_t^{\log \text{std}} &\sim \mathcal{GP}(0, K) \\
\text{obs}_t &\sim \text{Normal}(\text{return}, \exp(\log \text{std} + \Delta_t^{\log \text{std}}))
\end{aligned}
$$

The Model

I did the modelling choice to make it as simple as possible

$$
\begin{aligned}
\text{return} &\sim \text{Normal}(0, 2^{\frac{1}{250}} - 1) \\
\log \text{std} &\sim \text{Normal}(\log(2^{\frac{1}{250}} - 1), 0.05) \\
\text{ls} &\sim \text{Gamma}(30, 5) \\
{\color{red}\alpha_{\text{vol}}} &{\color{red}\sim \text{Exponential}(100)} \\
K(*, *) &= \alpha_{\text{vol}} \, \text{Matern32}(\text{ls}) \\
\Delta_t^{\log \text{std}} &\sim \mathcal{GP}(0, K) \\
\text{obs}_t &\sim \text{Normal}(\text{return}, \exp(\log \text{std} + \Delta_t^{\log \text{std}}))
\end{aligned}
$$

The Model

I did the modelling choice to make it as simple as possible

$$
\begin{aligned}
\text{return} &\sim \text{Normal}(0, 2^{\frac{1}{250}} - 1) \\
\log\text{std} &\sim \text{Normal}(\log(2^{\frac{1}{250}} - 1), 0.05) \\
\text{ls} &\sim \text{Gamma}(30, 5) \\
\alpha_{\text{vol}} &\sim \text{Exponential}(100) \\
K(*, *) &= \alpha_{\text{vol}}\,\text{Matern32}(\text{ls}) \\
\Delta_t^{\log\text{std}} &\sim \mathcal{GP}(0, K) \\
\text{obs}_t &\sim \text{Normal}(\text{return}, \exp(\log\text{std} + \Delta_t^{\log\text{std}}))
\end{aligned}
$$

The Model

I did the modelling choice to make it as simple as possible

$$
\begin{aligned}
\text{return} &\sim \text{Normal}(0, 2^{\frac{1}{250}} - 1) \\
\log \text{std} &\sim \text{Normal}(\log(2^{\frac{1}{250}} - 1), 0.05) \\
\text{ls} &\sim \text{Gamma}(30, 5) \\
\alpha_{\text{vol}} &\sim \text{Exponential}(100) \\
K(*, *) &= \alpha_{\text{vol}}\, \text{Matern32}(\text{ls}) \\
\color{red}{\Delta_t^{\log \text{std}}} &\color{red}{\sim \mathcal{GP}(0, K)} \\
\text{obs}_t &\sim \text{Normal}(\text{return}, \exp(\log \text{std} + \Delta_t^{\log \text{std}}))
\end{aligned}
$$

The Model

I did the modelling choice to make it as simple as possible

$$
\begin{aligned}
\text{return} &\sim \text{Normal}(0, 2^{\frac{1}{250}} - 1) \\
\log \text{std} &\sim \text{Normal}(\log(2^{\frac{1}{250}} - 1), 0.05) \\
\text{ls} &\sim \text{Gamma}(30, 5) \\
\alpha_{\text{vol}} &\sim \text{Exponential}(100) \\
K(*, *) &= \alpha_{\text{vol}} \, \text{Matern32}(\text{ls}) \\
\Delta_t^{\log \text{std}} &\sim \mathcal{GP}(0, K) \\
\text{obs}_t &\sim \text{Normal}(\text{return}, \exp(\log \text{std} + \Delta_t^{\log \text{std}}))
\end{aligned}
$$

Make the model faster!

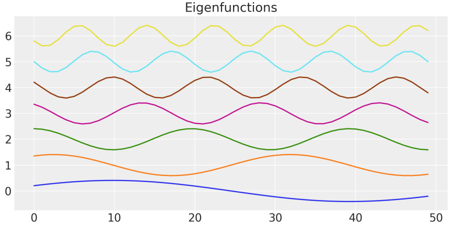
HSGP

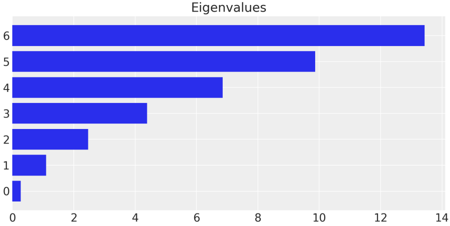
What is HSGP?

HSGP [1] is a short of **H**ilbert **S**pace **G**aussian **P**rocess.

Gentlemen FAQ:

- Approximates in function space
- Works for Stationary kernels
- Periodic Kernels are yet tricky
- Implemented in PyMC





Why HSGP?

1. Time series models have a lot of time-points
2. Algorithm complexity grows very fast ($\mathcal{O}(n^3)$)
3. HSGP [1] reduces the complexity to just $\mathcal{O}(mn + m)$
    - m - number of basis functions)

Gaussian process are now fast!

From $\mathcal{O}(n^3)$ to $\mathcal{O}(mn + m)$ makes so much sense!

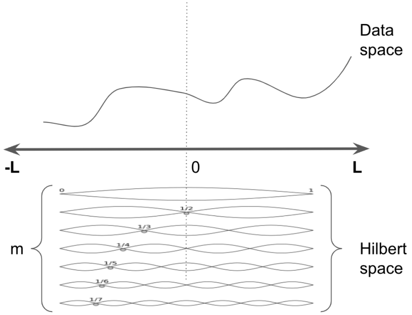
How to HSGP

There are two main parameters:

- L - extrapolation boundary
- m - number of basis functions

You can use it

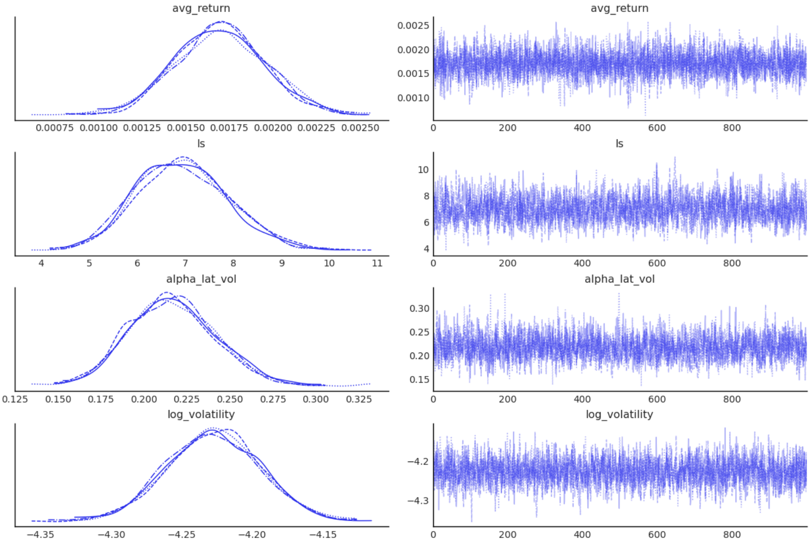
No need to be a mathematician to set L or m, there are good defaults in documentation.



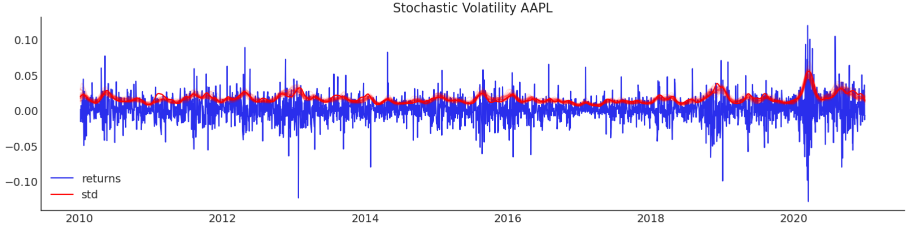
HSGP for Stochastic volatility

$$
\begin{aligned}
\text{return} &\sim \text{Normal}(0, 2^{\frac{1}{250}} - 1) \\
\log \text{std} &\sim \text{Normal}(2^{\frac{1}{250}} - 1, 0.05) \\
\text{ls} &\sim \text{Gamma}(30, 5) \\
\alpha_{\text{vol}} &\sim \text{Exponential}(100) \\
K(*, *) &= \alpha_{\text{vol}}\, \text{Matern32}(\text{ls}) \\
\Delta_t^{\log \text{std}} &\sim \mathcal{HSGP}(0, K) \\
\text{obs}_t &\sim \text{Normal}(\text{return}, \exp(\log \text{std} + \Delta_t^{\log \text{std}}))
\end{aligned}
$$

Results


avg_return
0.00075 0.00100 0.00125 0.00150 0.00175 0.00200 0.00225 0.00250
avg_return
0.0025
0.0020
0.0015
0.0010
0 200 400 600 800
ls
4 5 6 7 8 9 10 11
ls
10
8
6
4
0 200 400 600 800
alpha_lat_vol
0.125 0.150 0.175 0.200 0.225 0.250 0.275 0.300 0.325
alpha_lat_vol
0.30
0.25
0.20
0.15
0 200 400 600 800
log_volatility
−4.35 −4.30 −4.25 −4.20 −4.15
log_volatility
−4.2
−4.3
0 200 400 600 800

Takeaways



- Interpretable parameters
- Splines make model much faster
- Uncertainties for
  - average volatility
  - average returns
  - stochastic variation (α_{vol})

References I

G. Riutort-Mayol, P.-C. Bürkner, M. R. Andersen, A. Solin, and A. Vehtari.
Practical hilbert space approximate bayesian gaussian processes for probabilistic programming, 2022.

T. SJ and L. B.
Forecasting at scale.
2017.