State Space Gaussian Processes with Non-Gaussian Likelihoods

Hannes Nickisch[1] Arno Solin[2] Alexander Grigorievskiy[2,3]

[1]Philips Research, [2]Aalto University, [3]Silo.AI

ICML2018
July 13, 2018

Aalto University

SILO.AI

Outline

Gaussian Processes

Temporal GPs as stochastic differential equations (SDEs)

Learning and inference with Gaussian Likelihoods

Speeding up computation of state space model parameters

Non-Gaussian likelihoods

Approximate inference algorithms

Computational primitives and how to compute them

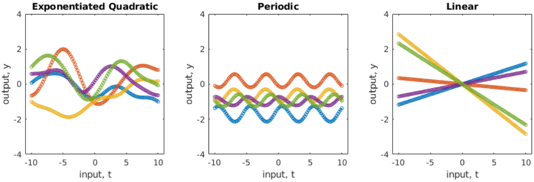
Experiments

Gaussian Processes (GPs)

Def:

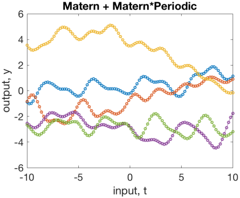
Gaussian Process (GP) is a stochastic process where for any inputs $\mathbf{t}$ all corresponding outputs $\boldsymbol{y}$ are distributed as $\boldsymbol{y} \sim \mathcal{N}(\mathbf{m}(\mathbf{t}), K(\boldsymbol{t}, \boldsymbol{t}|\theta))$. Denoted: $f(t) \sim \mathcal{GP}(m(t), k(t, t'|\theta))$

- Used as a prior over continuous functions in statistical models
- Properties (e.g. smoothness) are determined by the covariance function $k(t, t'|\theta)$



Temporal Gaussian Processes

- Input data is 1-D, usually time
- Fully probabilistic (Bayesian) approach
- Conveniently combining structural components by covariance operations



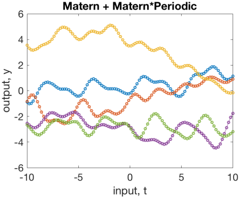
- Applicability for unevenly sampled data

Challenges:

- Large datasets
- Non-Gaussian likelihoods

Temporal Gaussian Processes

- Input data is 1-D, usually time
- Fully probabilistic (Bayesian) approach
- Conveniently combining structural components by covariance operations



- Applicability for unevenly sampled data

Challenges:

- Large datasets
- Non-Gaussian likelihoods

GP as a Stochastic Differential Equation (SDE)

Addressing challenge 1

Given a 1-D time series: $\{y_i, t_i\}_{i=1}^{N}$

- Gaussian Process model:

$$f(t) \sim \mathcal{GP}(m(t), k(t,t')) \quad \text{GP prior}$$

$$\mathbf{y} \,|\, \mathbf{f} \sim \prod_{i=1}^{n} \mathbb{P}(y_i \,|\, f(t_i)) \quad \text{Likelihood}$$

- Latent Posterior:

$$\mathbb{Q}(\mathbf{f} \,|\, \mathcal{D}) = \mathcal{N}\left(\mathbf{f} \,\middle|\, \mathbf{m} + \mathbf{K}\alpha, (\mathbf{K}^{-1} + \mathbf{W})^{-1}\right)$$

- Equivalent Stochastic Differential Equation (SDE) [3]

$$\frac{d\,\mathbf{f}(t)}{d\,t} = \mathbf{F}\mathbf{f}(t) + \mathbf{L}\mathbf{w}(t); \;\; \mathbf{f}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_\infty)$$

$$\mathbf{y} \,|\, \mathbf{f} \sim \prod_{i=1}^{n} \mathbb{P}(y_i \,|\, \mathbf{H}\mathbf{f}(t_i))$$

- $f(t) = \mathbf{H}\mathbf{f}(t)$
- $\mathbf{w}(t)$ - multidimensional white noise
- $\mathbf{F}, \mathbf{L}, \mathbf{H}, \mathbf{P}_\infty$ are determined from the covariance $\mathbf{K}$ [3]

GP as a Stochastic Differential Equation (SDE)

Addressing challenge 1

Given a 1-D time series: $\{y_i, t_i\}_{i=1}^{N}$

- Gaussian Process model:

$$f(t) \sim \mathcal{GP}(m(t), k(t, t')) \qquad \text{GP prior}$$

$$\mathbf{y} \,|\, \mathbf{f} \sim \prod_{i=1}^{n} \mathbb{P}(y_i \,|\, f(t_i)) \qquad \text{Likelihood}$$

- Latent Posterior:

$$\mathbb{Q}(\mathbf{f} \,|\, \mathcal{D}) = \mathcal{N}\left(\mathbf{f} \,\middle|\, \mathbf{m} + \mathbf{K}\boldsymbol{\alpha}, (\mathbf{K}^{-1} + \mathbf{W})^{-1}\right)$$

- Equivalent Stochastic Differential Equation (SDE) [3]

$$\frac{d\,\mathbf{f}(t)}{d\,t} = \mathbf{F}\mathbf{f}(t) + \mathbf{L}\mathbf{w}(t); \;\; \mathbf{f}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_\infty)$$

$$\mathbf{y} \,|\, \mathbf{f} \sim \prod_{i=1}^{n} \mathbb{P}(y_i \,|\, \mathbf{H}\mathbf{f}(t_i))$$

- $f(t) = \mathbf{H}\mathbf{f}(t)$
- $\mathbf{w}(t)$ - multidimensional white noise
- $\mathbf{F}, \mathbf{L}, \mathbf{H}, \mathbf{P}_\infty$ are determined from the covariance $\mathbf{K}$ [3]

Inference and Learning with Gaussian likelihood

Gaussian likelihood: $\mathbb{P}(y_i \mid f(t_i)) = \mathcal{N}(y_i \mid f(t_i), \sigma_n^2 \boldsymbol{I})$

- Posterior parameters:

$$\mathbf{W} = \sigma^{-2}\mathbf{I}_\mathrm{n}$$
$$\boldsymbol{\alpha} = (\mathbf{K} + \mathbf{W}^{-1})^{-1}(\mathbf{y} - \mathbf{m})$$

- Evidence:

$$\log Z_{\mathrm{GPR}} = -\frac{1}{2}\boldsymbol{\alpha}^\top(\mathbf{y} - \mathbf{m}) - \frac{1}{2}\log|\mathbf{K} + \mathbf{W}^{-1}| - \frac{N}{2}\log(2\pi\sigma_\mathrm{n}^2)$$

- The naïve approach has $\mathcal{O}(N^3)$ complexity

- Solve SDE between time points (equivalent discrete time model):

$$\mathbf{f}_i = \mathbf{A}_{i-1}\mathbf{f}_{i-1} + \mathbf{q}_{i-1}; \quad \mathbf{q}_{i-1} \sim \mathcal{N}(0, \mathbf{Q}_{i-1})$$
$$y_i = \mathbf{H}\mathbf{f}_\mathrm{i} + \epsilon_i; \quad \epsilon_n \sim \mathcal{N}(0, \sigma_n^2)$$

- Parameters of the discrete model:

$$\mathbf{A}_i = \mathbf{A}[\Delta t_i] = e^{\Delta t_i \mathbf{F}},$$
$$\mathbf{Q}_i = \mathbf{P}_\infty - \mathbf{A}_i \mathbf{P}_\infty \mathbf{A}_i^\top$$

- Inference and learning by Kalman FIlter (KF) and Rauch-Tung-Striebel (RTS) smoother in $\mathcal{O}(N)$ complexity

Inference and Learning with Gaussian likelihood

Gaussian likelihood: $\mathbb{P}(y_i \mid f(t_i)) = \mathcal{N}(y_i \mid f(t_i), \sigma_n^2 \boldsymbol{I})$

- Posterior parameters:

$$\mathbf{W} = \sigma^{-2}\mathbf{I}_{\mathrm{n}}$$
$$\boldsymbol{\alpha} = (\mathbf{K} + \mathbf{W}^{-1})^{-1}(\mathbf{y} - \mathbf{m})$$

- Evidence:

$$\log Z_{\mathrm{GPR}} = -\frac{1}{2}\boldsymbol{\alpha}^\top(\mathbf{y} - \mathbf{m}) - \frac{1}{2}\log|\mathbf{K} + \mathbf{W}^{-1}| - \frac{N}{2}\log(2\pi\sigma_{\mathrm{n}}^2)$$

- The naïve approach has $\mathcal{O}(N^3)$ complexity

- Solve SDE between time points (equivalent discrete time model):

$$\mathbf{f}_i = \mathbf{A}_{i-1}\mathbf{f}_{i-1} + \mathbf{q}_{i-1}; \quad \mathbf{q}_{i-1} \sim \mathcal{N}(0, \mathbf{Q}_{i-1})$$
$$y_i = \mathbf{H}\mathbf{f_i} + \epsilon_i; \quad \epsilon_n \sim \mathcal{N}(0, \sigma_n^2)$$

- Parameters of the discrete model:

$$\mathbf{A}_i = \mathbf{A}[\Delta t_i] = e^{\Delta t_i \mathbf{F}},$$
$$\mathbf{Q}_i = \mathbf{P}_\infty - \mathbf{A}_i \mathbf{P}_\infty \mathbf{A}_i^\top$$

- Inference and learning by Kalman FIlter (KF) and Rauch-Tung-Striebel (RTS) smoother in $\mathcal{O}(N)$ complexity

Fast computation of $\mathbf{A}_i$ and $\mathbf{Q}_i$ by interpolation

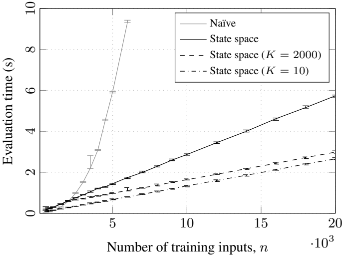
Problem:

- When there are many Δt_i parameters computation can be slow

Solution:

- $\psi : s \mapsto e^{s\mathbf{X}}$ is smooth mapping, hence interpolation (similar to KISS-GP [4])
- Evaluate ψ on an equispaced grid $s_1, s_2, .., s_K$, where $s_j = s_0 + j \cdot \Delta s$

- Use 4-point interpolation: $\mathbf{A} \approx c_1\mathbf{A}_{j-1} + c_2\mathbf{A}_j + c_3\mathbf{A}_{j+1} + c_4\mathbf{A}_{j+2}$. Coefficients $\{c_i\}_{i=1}^4$ are efficiently computable



Non-Gaussian Likelihoods

Addressing challenge 2

Posterior as a Gaussian approximation:

$$\mathbb{Q}(\mathbf{f}\,|\,\mathcal{D}) = \mathcal{N}\left(\mathbf{f}\,\middle|\,\mathbf{m} + \mathbf{K}\boldsymbol{\alpha}, (\mathbf{K}^{-1} + \mathbf{W})^{-1}\right)$$

- Laplace approximation (LA)
- Variational Bayes (VB)
- Direct Kullback-Liebler minimization (KL)
- Assumed Density Filtering (ADF) a.k.a. single sweep Expectation Propagation (EP)

Laplace Approximation

- $\log \mathbb{P}(\mathbf{f}\,|\,\mathcal{D}) \sim \log \mathbb{P}(\mathbf{f}\,|\,\mathbf{y}) + \log \mathbb{P}(\mathbf{f}\,|\,\mathbf{t})$
- Find the mode $\hat{\mathbf{f}}$ of this function by Newton method
- Hessian at the mode $\hat{\mathbf{f}}$ is precision $\mathbf{W} = -\partial^2 \log \mathbb{P}(\hat{\mathbf{f}}\,|\,\mathbf{t})$
- $\log Z_{LA} = -\frac{1}{2}\left[\boldsymbol{\alpha}^\top \mathrm{mvm}_{\mathbf{K}}(\boldsymbol{\alpha}) + \mathrm{ld}_{\mathbf{K}}(\mathbf{W}) - 2\sum_i \log \mathbb{P}(y_i|\hat{f}_i)\right]$

Computational Primitives

The following computational primitives allow to cast the covariance approximation in more generic terms:

- Linear system solving: $\text{solve}_{\mathbf{K}}(\mathbf{W}, \mathbf{r}) := (\mathbf{K} + \mathbf{W}^{-1})^{-1}\mathbf{r}$
- Matrix-vector multiplications: $\text{mvm}_{\mathbf{K}}(\mathbf{r}) := \mathbf{K}\mathbf{r}$
- Log-determinants: $\text{ld}_{\mathbf{K}}(\mathbf{W}) := \log|\mathbf{B}|$ with well-conditioned $\mathbf{B} = \mathbf{I} + \mathbf{W}^{\frac{1}{2}}\,\mathbf{K}\,\mathbf{W}^{\frac{1}{2}}$
- Predictions need latent mean $\mathbb{E}[f_*]$ and variance $\mathbb{V}[f_*]$

Tackling computational primitives

Using state space from of temporal GPs

SpInGP:

- The first two computational primitives are calculated using *SpInGP* [5] approach:
- Idea is: using state space form compose the inverse of the covariance matrix, which turns out to be block-tridiagonal

KF and RTS Smoothing:

- The last two primitives are solved by Kalman filtering and RTS smoothing
- Predictions are computed by primitive 4 and then by propagation through likelihood

Comments:

- Derivatives of computational primitives, required for learning, are computed in a similar way
- *SpInGP* involves computations with block-tridiagonal matrices. These computations are similar to KF and RTS smoothing (see [1] Appendix)

Experiments 2-3

Experiments are designed to emphasize the paper findings and statements

1. A robust regression (Student's *t* likelihood) study example with $n = 34{,}154$ observations
2. Numerical effects in non-Gaussian likelihoods

Table 1. A representative subset of supported likelihoods and inference schemes (for a full list, see Rasmussen & Nickisch, 2010). Results for simulated data with $n = 1000$ (around the break-even point of computational benefits). Results compared to respective naïve solution in mean absolute error (MAE). †The results for EP are compared against ADF explaining the deviation and speed-up.

Likelihood	Inference	MAE in $\boldsymbol{\alpha}$	MAE in $\mathbf{W}$	MAE in $\boldsymbol{\mu}_{f,*}$	$-\log Z$	$-\log Z_{\text{ss}}$	t/t_{ss}	Description
Gaussian	Exact	$< 10^{-4}$	$< 10^{-16}$	$< 10^{-14}$	-1252.29	-1252.30	2.0	Regression
Student's t	Laplace	$< 10^{-7}$	$< 10^{-6}$	$< 10^{-3}$	2114.45	2114.45	1.4	Regression,
Student's t	VB	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-7}$	2114.72	2114.72	2.7	robust
Student's t	KL	$< 10^{-4}$	$< 10^{-4}$	$< 10^{-5}$	2114.86	2114.86	4.6	
Poisson	Laplace	$< 10^{-6}$	$< 10^{-4}$	$< 10^{-6}$	1200.11	1200.11	1.2	Poisson regression,
Poisson	EP/ADF†	$< 10^{-1}$	$< 10^{0}$	$< 10^{-2}$	1200.11	1206.59	39.5	count data
Logistic	Laplace	$< 10^{-8}$	$< 10^{-7}$	$< 10^{-7}$	491.58	491.58	1.3	Classification,
Logistic	VB	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	492.36	492.36	2.3	logit regression
Logistic	KL	$< 10^{-7}$	$< 10^{-6}$	$< 10^{-7}$	491.57	491.57	4.0	
Logistic	EP/ADF†	$< 10^{-1}$	$< 10^{0}$	$< 10^{-1}$	491.50	525.46	48.1	
Erf	Laplace	$< 10^{-8}$	$< 10^{-6}$	$< 10^{-7}$	392.01	392.01	1.2	Classification,
Erf	EP/ADF†	$< 10^{0}$	$< 10^{0}$	$< 10^{-1}$	392.01	433.75	37.1	probit regression

Experiment 4

- A new interesting data set with commercial airline accidents dates scraped from Wikipedia [6]
- Accidents over the time-span of $\sim$100 years, $n = 35{,}959$ days
- We model the accident intensity as a Log Gaussian Cox process (Poisson likelihood)
- The GP prior is set up as:

$$k(t, t') = k_{\text{Mat.}}(t, t') + k_{\text{per.}}(t, t')\, k_{\text{Mat.}}(t, t')$$

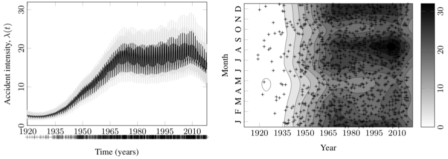


Figure 2: (a) Intensity of aircraft incidents modeled by a log Gaussian Cox process with the mean and approximate 90% confidence regions visualized ($N = 35{,}959$). **(b)** The time course of the seasonal effect in the airline accident intensity, plotted in a year vs. month plot (with wrap-around continuity between edges).

Conclusions

- This paper brings together research done in state space GPs and non-Gaussian approximate inference
- We improve stability and provide additional speed-up by fast computations of the state space model parameters
- We provide unifying code for all approches in GPML toolbox v. 4.2 [7]
- Visit our poster: #151

References

[1] H. Nickisch, A. Solin, and A. Grigorievskiy (2018). State Space Gaussian Processes with Non-Gaussian Likelihood. In *ICML*.

[2] C.E. Rasmussen and C.K.I. Williams (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

[3] J. Hartikainen, and S. Särkkä (2010). Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *MLSP*.

[4] A. G. Wilson, and H. Nickisch (2015). Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP). In *ICML*.

[5] A. Grigorievskiy, N. Lawrence, and S. Särkkä (2017). Parallelizable Sparse Inverse Formulation Gaussian Processes (SpInGP). In *MLSP*.

[6] Wikipedia (2018). URL `https://en.wikipedia.org/wiki/List_of_accidents_and_incidents_involving_commercial_aircraft`

[7] C. E. Rasmussen and H. Nickisch (2010). Gaussian Processes for Machine Learning (GPML). In *GMLR*.