Machine Learning with Signal Processing
Part III: Three Views into Gaussian Processes

Arno Solin
Assistant Professor in Machine Learning
Department of Computer Science
Aalto University

ICML 2020 Tutorial

@arnosolin  arno.solin.fi
Outline

- GPs for long data
- Kernel representation
- Spectral representation
- State space representation
- Connection to sparse precision
- Linear-time GP regression
- Connection to sparse precision
GPs for long (unbounded) data

Three views into GPs

Kernel (moment)

Spectral (Fourier)

State space (path)
Kernel (moment) representation

\[ f(t) \sim \text{GP}(\mu(t), \kappa(t, t')) \quad \text{GP prior} \]

\[ y \mid f \sim \prod_i p(y_i \mid f(t_i)) \quad \text{likelihood} \]

- Let’s focus on the GP prior only.
- A temporal Gaussian process (GP) is a random function \( f(t) \), such that joint distribution of \( f(t_1), \ldots, f(t_n) \) is always Gaussian.
- Mean and covariance functions have the form:
  \[
  \mu(t) = \mathbb{E}[f(t)], \\
  \kappa(t, t') = \mathbb{E}[(f(t) - \mu(t))(f(t') - \mu(t'))^T].
  \]
- Convenient for model specification, but expanding the kernel to a covariance matrix can be problematic (the notorious \( \mathcal{O}(n^3) \) scaling).
Spectral (Fourier) representation

- The Fourier transform of a function \( f(t) : \mathbb{R} \rightarrow \mathbb{R} \) is

\[
\mathcal{F}[f](i \omega) = \int_{\mathbb{R}} f(t) \exp(-i \omega t) \, dt
\]

- For a stationary GP, the covariance function can be written in terms of the difference between two inputs:

\[
\kappa(t, t') \triangleq \kappa(t - t')
\]

- Wiener–Khinchin: If \( f(t) \) is a stationary Gaussian process with covariance function \( \kappa(t) \), then its spectral density is \( S(\omega) = \mathcal{F}[\kappa] \).

- Spectral representation of a GP in terms of spectral density function

\[
S(\omega) = \mathbb{E}[\tilde{f}(i \omega) \tilde{f}^T(-i \omega)]
\]
State space (path) representation [1/3]

► Path or state space representation as solution to a linear time-invariant (LTI) stochastic differential equation (SDE):

\[ df = F f dt + L d\beta, \]

where \( f = (f, df/dt, \ldots) \) and \( \beta(t) \) is a vector of Wiener processes.

► Equivalently, but more informally

\[ \frac{df(t)}{dt} = F f(t) + L w(t), \]

where \( w(t) \) is white noise.

► The model now consists of a drift matrix \( F \in \mathbb{R}^{m \times m} \), a diffusion matrix \( L \in \mathbb{R}^{m \times s} \), and the spectral density matrix of the white noise process \( Q_c \in \mathbb{R}^{s \times s} \).

► The scalar-valued GP can be recovered by \( f(t) = h^T f(t) \).
The initial state is given by a stationary state $f(0) \sim N(0, P_\infty)$ which fulfills

$$FP_\infty + P_\infty F^T + LQ_cL^T = 0$$

The covariance function at the stationary state can be recovered by

$$\kappa(t, t') = \begin{cases} h^T P_\infty \exp((t' - t)F)^T h, & t' \geq t \\ h^T \exp((t' - t)F) P_\infty h, & t' < t \end{cases}$$

where $\exp(\cdot)$ denotes the matrix exponential function.

The spectral density function at the stationary state can be recovered by

$$S(\omega) = h^T (F + i \omega I)^{-1} LQ_c L^T (F - i \omega I)^{-T} h$$
State space (path) representation [3/3]

- Similarly as the kernel has to be evaluated into a covariance matrix for computations, the SDE can be solved for discrete time points \( \{t_i\}_{i=1}^n \).
- The resulting model is a discrete state space model:
  \[
  f_i = A_i f_{i-1} + q_{i-1}, \quad q_i \sim \mathcal{N}(0, Q_i),
  \]
  where \( f_i = f(t_i) \).

- The discrete-time model matrices are given by:
  \[
  A_i = \exp(F \Delta t_i),
  Q_i = \int_0^{\Delta t_i} \exp(F (\Delta t_i - \tau)) L Q_c L^T \exp(F (\Delta t_i - \tau))^T d\tau,
  \]
  where \( \Delta t_i = t_{i+1} - t_i \).

- If the model is stationary, \( Q_i \) is given by
  \[
  Q_i = P_\infty - A_i P_\infty A_i^T
  \]
Three views into GPs

Covariance function

\[ \tau = t - t' \]

Spectral density function

Sample functions
Example: Exponential covariance function

- Exponential covariance function (Ornstein-Uhlenbeck process):
  \[
  \kappa(t, t') = \exp(-\lambda |t - t'|)
  \]

- Spectral density function:
  \[
  S(\omega) = \frac{2}{\lambda + \omega^2/\lambda}
  \]

- Path representation: Stochastic differential equation (SDE)
  \[
  \frac{df(t)}{dt} = -\lambda f(t) + w(t),
  \]
  or using the notation from before:
  \[
  F = -\lambda, \quad L = 1, \quad Q_c = 2, \quad h = 1, \quad \text{and} \quad P_\infty = 1.
  \]
### Examples of applicable GP priors

<table>
<thead>
<tr>
<th>Priors</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td><img src="image" alt="Constant" /></td>
</tr>
<tr>
<td>Linear</td>
<td><img src="image" alt="Linear" /></td>
</tr>
<tr>
<td>Wiener process</td>
<td><img src="image" alt="Wiener process" /></td>
</tr>
<tr>
<td>Wiener velocity</td>
<td><img src="image" alt="Wiener velocity" /></td>
</tr>
<tr>
<td>Exponential</td>
<td><img src="image" alt="Exponential" /></td>
</tr>
<tr>
<td>Matérn ($\nu = 3/2$)</td>
<td>![Matérn ($\nu = 3/2$)]</td>
</tr>
<tr>
<td>Matérn ($\nu = 5/2$)</td>
<td>![Matérn ($\nu = 5/2$)]</td>
</tr>
<tr>
<td>Squared exponential</td>
<td><img src="image" alt="Squared exponential" /></td>
</tr>
<tr>
<td>Rational quadratic</td>
<td><img src="image" alt="Rational quadratic" /></td>
</tr>
<tr>
<td>Periodic</td>
<td><img src="image" alt="Periodic" /></td>
</tr>
<tr>
<td>Quasi-periodic</td>
<td><img src="image" alt="Quasi-periodic" /></td>
</tr>
<tr>
<td>Noise</td>
<td><img src="image" alt="Noise" /></td>
</tr>
</tbody>
</table>
Applicable GP priors

- The covariance function needs to be Markovian (or approximated as such).
- Covers many common stationary and non-stationary models.

- **Sums of kernels**: $\kappa(t, t') = \kappa_1(t, t') + \kappa_2(t, t')$
  - Stacking of the state spaces
  - State dimension: $m = m_1 + m_2$

- **Product of kernels**: $\kappa(t, t') = \kappa_1(t, t') \kappa_2(t, t')$
  - Kronecker sum of the models
  - State dimension: $m = m_1 m_2$
Example: GP regression, $O(n^3)$
Consider the GP regression problem with input–output training pairs \( \{(t_i, y_i)\}_{i=1}^{n} \):

\[
\begin{align*}
  f(t) &\sim \text{GP}(0, \kappa(t, t')) , \\
  y_i &= f(t_i) + \varepsilon_i, \quad \varepsilon_i \sim \text{N}(0, \sigma_n^2)
\end{align*}
\]

The posterior mean and variance for an unseen test input \( t_* \) is given by (see previous lectures):

\[
\begin{align*}
  \mathbb{E}[f_*] &= k_* (K + \sigma_n^2 I)^{-1} y, \\
  \mathbb{V}[f_*] &= K_{**} - k_* (K + \sigma_n^2 I)^{-1} k_*^T
\end{align*}
\]

Note the inversion of the \( n \times n \) matrix.
Example: GP regression, $\mathcal{O}(n^3)$
Example: GP regression, $O(n)$

- The sequential solution (goes under the name ‘Kalman filter’) considers one data point at a time, hence the linear time-scaling.
- Start from $m_0 = 0$ and $P_0 = P_\infty$ and for each data point iterate the following steps.
- Kalman prediction:
  
  $m_{i|i-1} = A_{i-1} m_{i-1|i-1},$
  
  $P_{i|i-1} = A_{i-1} P_{i-1|i-1} A_{i-1}^T + Q_{i-1}.$

- Kalman update:
  
  $v_i = y_i - h^T m_{i|i-1},$
  
  $S_i = h^T P_{i|i-1} h + \sigma_n^2,$
  
  $K_i = P_{i|i-1} h S_i^{-1},$
  
  $m_{i|i} = m_{i|i-1} + K_i v_i,$
  
  $P_{i|i} = P_{i|i-1} - K_i S_i K_i^T.$
Example: GP regression, $O(n)$

- To condition all time-marginals on all data, run a backward sweep (Rauch–Tung–Striebel smoother):

$$
\begin{align*}
\mathbf{m}_{i+1|i} &= \mathbf{A}_i \mathbf{m}_{i|i}, \\
\mathbf{P}_{i+1|i} &= \mathbf{A}_i \mathbf{P}_{i|i} \mathbf{A}_i^T + \mathbf{Q}_i, \\
\mathbf{G}_i &= \mathbf{P}_{i|i} \mathbf{A}_i^T \mathbf{P}_{i+1|i}^{-1}, \\
\mathbf{m}_{i|i} &= \mathbf{m}_{i|i} + \mathbf{G}_i (\mathbf{m}_{i+1|i} - \mathbf{m}_{i+1|i}), \\
\mathbf{P}_{i|i} &= \mathbf{P}_{i|i} + \mathbf{G}_i (\mathbf{P}_{i+1|i} - \mathbf{P}_{i+1|i}) \mathbf{G}_i^T.
\end{align*}
$$

- The marginal mean and variance can be recovered by:

$$
\begin{align*}
\mathbb{E}[f_i] &= \mathbf{h}^T \mathbf{m}_{i|i}, \\
\mathbb{V}[f_i] &= \mathbf{h}^T \mathbf{P}_{i|i} \mathbf{h}
\end{align*}
$$

- The log marginal likelihood can be evaluated as a by-product of the Kalman update:

$$
\log p(\mathbf{y}) = -\frac{1}{2} \sum_{i=1}^{n} \log |2\pi S_i| + \mathbf{v}_i^T S_i^{-1} \mathbf{v}_i
$$
Example: GP regression, $\mathcal{O}(n)$
Basic regression example

- Number of births in the US (from BDA3 by Gelman et al.)
- Daily data between 1969–1988 ($n = 7305$)
- GP regression with a prior covariance function:

\[
\kappa(t, t') = \kappa_{\text{Mat.}}^{\nu=5/2}(t, t') + \kappa_{\text{Mat.}}^{\nu=3/2}(t, t') + \kappa_{\text{Per.}}^{\text{year}}(t, t') \kappa_{\text{Mat.}}^{\nu=3/2}(t, t') + \kappa_{\text{Per.}}^{\text{week}}(t, t') \kappa_{\text{Mat.}}^{\nu=3/2}(t, t')
\]

- Learn hyperparameters by optimizing the marginal likelihood
Basic regression example

- Number of births in the US (from BDA3 by Gelman et al.)
- Daily data between 1969–1988 (n = 7305)
- GP regression with a prior covariance function:
  \[ \kappa(t, t') = \kappa_{\text{Mat.}}(t, t') + \kappa_{\text{year Per.}}(t, t') + \kappa_{\text{week Per.}}(t, t') \]

Learn hyperparameters by optimizing the marginal likelihood

Explaining changes in number of births in the US
Connection to banded precision matrices
Precision matrices

Covariance (Gram) matrix:
\[ K = \kappa(X, X) \]

Precision matrix:
\[ K^{-1} \]

For Markovian models the precision is sparse!
(block tri-diagonal)

see Durrande et al. AISTATS 2019.
Constructing the precision matrix

The full precision matrix can be constructed from the state space model matrices:

\[
\hat{K}^{-1} = \left( \begin{array}{cccccc}
1 & 0 & 0 & \cdots & 0 \\
-A_1 & 1 & 0 & \cdots & 0 \\
0 & -A_2 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & -A_n & 1
\end{array} \right)^{-T} \left( \begin{array}{cccc}
P_0 & 0 & \cdots & 0 \\
0 & Q_1 & 0 & \cdots & 0 \\
0 & 0 & Q_2 & \cdots & \vdots \\
0 & 0 & \cdots & Q_n \\
0 & 0 & \cdots & 0 & \end{array} \right)^{-1} \left( \begin{array}{cccc}
1 & 0 & 0 & \cdots & 0 \\
-A_1 & 1 & 0 & \cdots & 0 \\
0 & -A_2 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & -A_n & 1
\end{array} \right)^{-1}
\]

Discarding the other model states by passing through the measurement model:

\[
K^{-1} = (I_n \otimes h) \hat{K}^{-1} (I_n \otimes h)^T
\]
Summary

- Gaussian processes have different representations:
  - Covariance function
  - Spectral density
  - State space

- Temporal (single-input) Gaussian processes
  $\iff$ stochastic differential equations (SDEs)

- Conversions between the representations can make model building easier

- (Exact) inference of the latent functions, can be done in $O(n)$ time and memory complexity by Kalman filtering
Up next

- Spatio-temporal models
- Non-Gaussian likelihood models
- Hybrid models
These references are sources for finding a more detailed overview on the topics of this part:
