Models and inference for temporal Gaussian processes

(*i.e.*, GPs for signal processing)

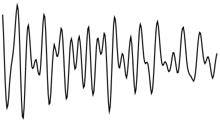
William Wilkinson

Postdoctoral researcher in machine learning
Department of computer science
Aalto University, Finland

Winter 2019

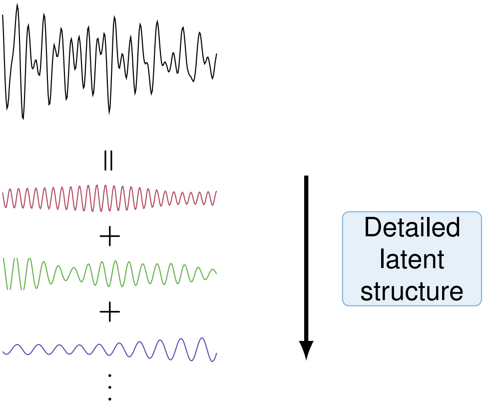
Problem domain

We are interested in applying Gaussian process models to time-domain signals.



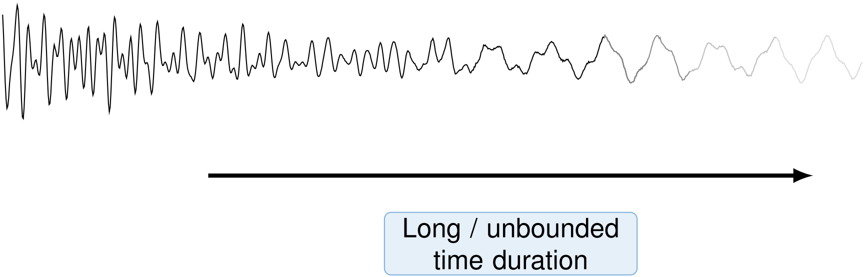
Problem domain

We are interested in applying Gaussian process models to time-domain signals.



Problem domain

We are interested in applying Gaussian process models to time-domain signals.



A contradiction

GPs for signal processing

Often acquire data at a rate of many ’000s per second

$\mathcal{O}(n^3)$ inference

A contradiction

GPs for signal processing

Often acquire data at a rate of many ’000s per second

$\mathcal{O}(n^3)$ inference

Sparse GPs are not a natural fit for signals:

- still $\sim$cubic scaling
- smoothing causes a loss of perceptually important information

Solution - State space models

To get around this contradiction, we exploit the sequential nature of our data to reformulate our model.

First, some setting up:

First, some setting up:

Think of signal processing as learning a function of time, allowing us extrapolate, interpolate, reduce noise, . . .

First, some setting up:

Think of signal processing as learning a function of time, allowing us extrapolate, interpolate, reduce noise, . . .

$$f(t) \sim \mathcal{GP}(0, K_\theta(t, t'))$$

First, some setting up:

Think of signal processing as learning a function of time, allowing us extrapolate, interpolate, reduce noise, ...

$$f(t) \sim \mathcal{GP}(0, K_\theta(t, t'))$$

A GP prior states:

- evaluations of $f(\cdot)$ are jointly Gaussian
- covariance between time steps is determined by $K_\theta(\cdot, \cdot)$

"The state of a dynamic system is the smallest collection of numbers which must be specified at time t_k in order to be able to predict the behaviour of the system for any time $t_{k+1} \geq t_k$."

Rudolf E. Kalman

$$f(t) \sim \mathcal{GP}(0, K_\theta(t, t'))$$

"The state of a dynamic system is the smallest collection of numbers which must be specified at time t_k in order to be able to predict the behaviour of the system for any time $t_{k+1} \geq t_k$."

Rudolf E. Kalman

$$f(t) \sim \mathcal{GP}(0, K_\theta(t, t'))$$

For a GP, the "numbers" required might be:

"The state of a dynamic system is the smallest collection of numbers which must be specified at time t_k in order to be able to predict the behaviour of the system for any time $t_{k+1} \geq t_k$."

Rudolf E. Kalman

$$f(t) \sim \mathcal{GP}(0, K_\theta(t, t'))$$

For a GP, the "numbers" required might be:

- some time derivatives, $\mathbf{f}_k = (f(t_k), \dot{f}(t_k), \ddot{f}(t_k), \ldots)^\top$

$$\mathbf{f}_{k+1} = \qquad \mathbf{f}_k$$

"The state of a dynamic system is the smallest collection of numbers which must be specified at time t_k in order to be able to predict the behaviour of the system for any time $t_{k+1} \geq t_k$."

Rudolf E. Kalman

$$f(t) \sim \mathcal{GP}(0, K_\theta(t, t'))$$

For a GP, the "numbers" required might be:

- some time derivatives, $\mathbf{f}_k = (f(t_k), \dot{f}(t_k), \ddot{f}(t_k), \ldots)^\top$
- a transition model between time steps, $\mathbf{A}_{\theta,k}$

$$\mathbf{f}_{k+1} = \mathbf{A}_{\theta,k}\mathbf{f}_k$$

"The state of a dynamic system is the smallest collection of numbers which must be specified at time t_k in order to be able to predict the behaviour of the system for any time $t_{k+1} \geq t_k$."

Rudolf E. Kalman

$$f(t) \sim \mathcal{GP}(0, K_\theta(t, t'))$$

For a GP, the "numbers" required might be:

- some time derivatives, $\mathbf{f}_k = (f(t_k), \dot{f}(t_k), \ddot{f}(t_k), \ldots)^\top$
- a transition model between time steps, $\mathbf{A}_{\theta,k}$
- a process noise covariance, $\mathbf{Q}_{\theta,k}$

$$\mathbf{f}_{k+1} = \mathbf{A}_{\theta,k}\mathbf{f}_k + \mathbf{q}_k, \qquad \mathbf{q}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}_{\theta,k})$$

"The state of a dynamic system is the smallest collection of numbers which must be specified at time t_k in order to be able to predict the behaviour of the system for any time $t_{k+1} \geq t_k$."

Rudolf E. Kalman

$$f(t) \sim \mathcal{GP}(0, K_\theta(t, t'))$$

For a GP, the "numbers" required might be:

- some time derivatives, $\mathbf{f}_k = (f(t_k), \dot{f}(t_k), \ddot{f}(t_k), \ldots)^\top$
- a transition model between time steps, $\mathbf{A}_{\theta,k}$
- a process noise covariance, $\mathbf{Q}_{\theta,k}$
- an observation model, $\mathbf{h}^\top = (1, 0, 0, \ldots)$

$$\mathbf{f}_{k+1} = \mathbf{A}_{\theta,k}\mathbf{f}_k + \mathbf{q}_k, \qquad \mathbf{q}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}_{\theta,k})$$
$$f(t_{k+1}) = \mathbf{h}^\top \mathbf{f}_{k+1}$$

Stochastic differential equations

$$\mathbf{f}_{k+1} = \mathbf{A}_{\theta,k}\mathbf{f}_k + \mathbf{q}_k, \qquad \mathbf{q}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}_{\theta,k})$$

This state space model is the discrete-time solution to the linear time-invariant (LTI) stochastic differential equation (SDE):

$$\frac{\mathrm{d}\mathbf{f}(t)}{\mathrm{d}t} = \mathbf{F}_\theta \mathbf{f}(t) + \mathbf{L}\mathbf{w}(t)$$

Stochastic differential equations

$$\mathbf{f}_{k+1} = \mathbf{A}_{\theta,k}\mathbf{f}_k + \mathbf{q}_k, \qquad \mathbf{q}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}_{\theta,k})$$

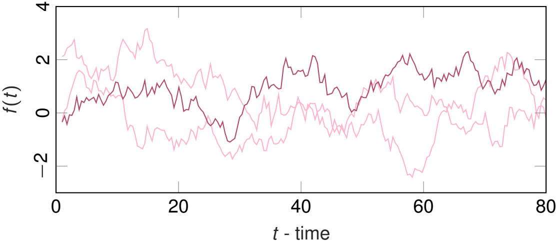
This state space model is the discrete-time solution to the linear time-invariant (LTI) stochastic differential equation (SDE):

$$\frac{\mathrm{d}\mathbf{f}(t)}{\mathrm{d}t} = \mathbf{F}_\theta \mathbf{f}(t) + \mathbf{L}\mathbf{w}(t)$$

with initial state $\mathbf{f}(t_0) = \mathrm{N}(\mathbf{0}, \mathbf{P}_\infty)$, for some stationary covariance $\mathbf{P}_\infty$.

- $\mathbf{w}(t)$ is white noise with spectral density $\mathbf{Q}_c$
- $\mathbf{F}_\theta$ is a *feedback* matrix, $\mathbf{A}_{\theta,k} = \exp(\mathbf{F}_\theta(t_{k'} - t_k))$
- $\mathbf{L}$ is a *noise-effect* matrix

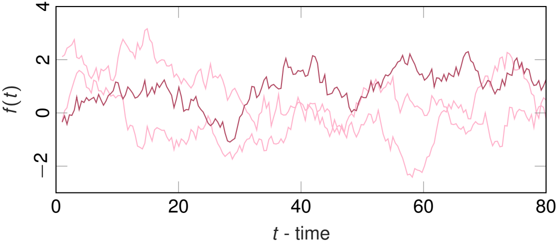
1D example



Samples from the Ornstein-Uhlenbeck process prior with $q = 0.2$, $\lambda = 0.1$.

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = -\lambda f(t) + w(t)$$

1D example

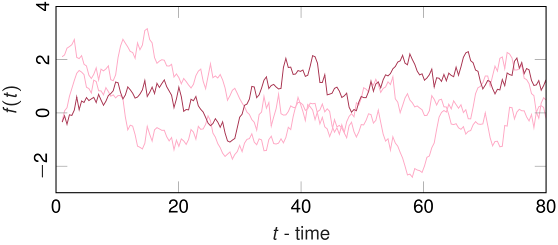


Samples from the Ornstein-Uhlenbeck process prior with $q = 0.2$, $\lambda = 0.1$.

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = -\lambda f(t) + w(t)$$

we have set $\mathbf{F}_\theta = -\lambda$, $\mathbf{L} = 1$, $\mathbf{Q}_c = q$, $\mathbf{h}^\top = 1 \implies \mathbf{P}_\infty = q/2\lambda$.

1D example



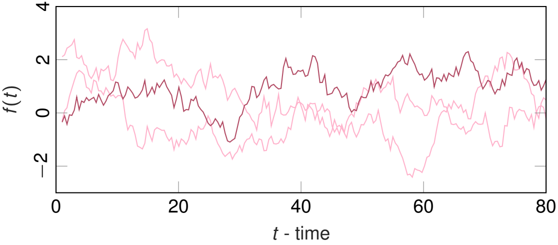
Samples from the Ornstein-Uhlenbeck process prior with $q = 0.2$, $\lambda = 0.1$.

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = -\lambda f(t) + w(t)$$

we have set $\mathbf{F}_\theta = -\lambda$, $\mathbf{L} = 1$, $\mathbf{Q}_c = q$, $\mathbf{h}^\top = 1 \implies \mathbf{P}_\infty = q/2\lambda$.

$$K(t, t') = \begin{cases} \mathbf{h}^\top \mathbf{P}_\infty \exp((t' - t)\mathbf{F}_\theta)^\top \mathbf{h}, & \text{if } t' - t \geq 0 \\ \mathbf{h}^\top \exp(-(t' - t)\mathbf{F}_\theta)\mathbf{P}_\infty \mathbf{h}, & \text{if } t' - t < 0 \end{cases}$$

1D example



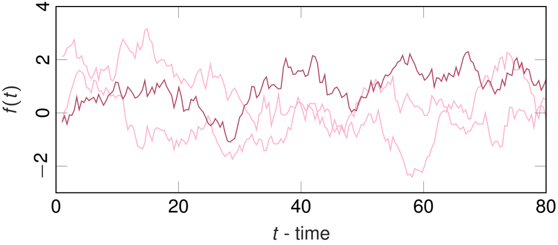
Samples from the Ornstein-Uhlenbeck process prior with $q = 0.2$, $\lambda = 0.1$.

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = -\lambda f(t) + w(t)$$

we have set $\mathbf{F}_\theta = -\lambda$, $\mathbf{L} = 1$, $\mathbf{Q}_c = q$, $\mathbf{h}^\top = 1 \implies \mathbf{P}_\infty = q/2\lambda$.

$$K(t, t') = \begin{cases} q/2\lambda \, \exp(-(t' - t)\lambda), & \text{if } t' - t \geq 0 \\ \exp((t' - t)\lambda) \, q/2\lambda, & \text{if } t' - t < 0 \end{cases}$$

1D example



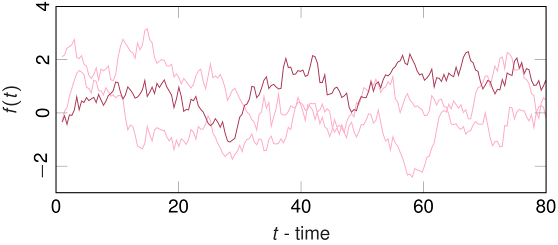
Samples from the Ornstein-Uhlenbeck process prior with $q = 0.2$, $\lambda = 0.1$.

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = -\lambda f(t) + w(t)$$

we have set $\mathbf{F}_\theta = -\lambda$, $\mathbf{L} = 1$, $\mathbf{Q}_c = q$, $\mathbf{h}^\top = 1 \implies \mathbf{P}_\infty = q/2\lambda$.

$$K(t, t') = q/2\lambda \exp(-|t' - t|\lambda)$$

1D example



Samples from the Ornstein-Uhlenbeck process prior with $q = 0.2$, $\lambda = 0.1$.

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = -\lambda f(t) + w(t)$$

we have set $\mathbf{F}_\theta = -\lambda$, $\mathbf{L} = 1$, $\mathbf{Q}_c = q$, $\mathbf{h}^\top = 1 \implies \mathbf{P}_\infty = q/2\lambda$.

$$K(t, t') = \sigma^2 \exp(-|t' - t|/\ell), \qquad \sigma^2 = q/2\lambda,\ \ell = 1/\lambda$$

There exists a dual kernel / SDE form for most popular GP models

$$f(t) \sim \mathcal{GP}\big(0,\, K_\theta(t,t')\big), \qquad y_k \sim \mathrm{N}(f(t_k), \sigma_y^2)$$

$$\mathbf{f}_k = \mathbf{A}_{\theta,k}\mathbf{f}_{k-1} + \mathbf{q}_{k-1}, \qquad y_k = \mathbf{h}^\top \mathbf{f}_k + \sigma_y \epsilon_k$$

There exists a dual kernel / SDE form for most popular GP models

$$f(t) \sim \mathcal{GP}\big(0,\, K_\theta(t,t')\big), \qquad \mathbf{f}_k = \mathbf{A}_{\theta,k}\mathbf{f}_{k-1} + \mathbf{q}_{k-1},$$
$$y_k \sim \mathrm{N}(f(t_k), \sigma_y^2) \qquad y_k = \mathbf{h}^\top \mathbf{f}_k + \sigma_y \epsilon_k$$

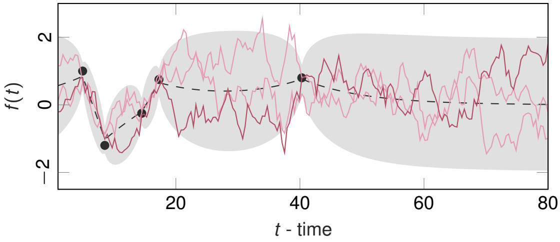
and inference is performed in $\mathcal{O}(nm^3)$ via Kalman filtering and smoothing ($n =$ # data, $m =$ dimensionality of $\mathbf{f}_k$)

There exists a dual kernel / SDE form for most popular GP models

$$f(t) \sim \mathcal{GP}\big(0,\, K_\theta(t,t')\big), \qquad \mathbf{f}_k = \mathbf{A}_{\theta,k}\mathbf{f}_{k-1} + \mathbf{q}_{k-1},$$

$$y_k \sim \mathrm{N}(f(t_k), \sigma_y^2) \qquad y_k = \mathbf{h}^\top \mathbf{f}_k + \sigma_y \epsilon_k$$

and inference is performed in $\mathcal{O}(nm^3)$ via Kalman filtering and smoothing ($n =$ # data, $m =$ dimensionality of $\mathbf{f}_k$)



Samples from the posterior distribution with $\sigma_y^2 = 0.05$.

Back to signal processing:

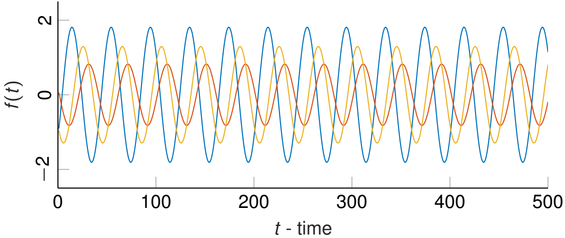
we are interested in models that capture periodic latent structure.

The cosine kernel

$$K(t, t) = \cos(\omega(t - t'))$$

The cosine kernel

$$K(t, t) = \cos(\omega(t - t'))$$



Sample paths are pure sinusoids, *i.e.*, there is no noise process

The cosine kernel

$$K(t,t) = \cos(\omega(t - t'))$$

state space representation is actually a complex ODE:

$$\begin{pmatrix} \mathrm{Re}[\dot{f}(t)] \\ \mathrm{Im}[\dot{f}(t)] \end{pmatrix} = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Re}[f(t)] \\ \mathrm{Im}[f(t)] \end{pmatrix}$$

The cosine kernel

$$K(t,t) = \cos(\omega(t - t'))$$

state space representation is actually a complex ODE:

$$\begin{pmatrix} \mathrm{Re}[\dot{f}(t)] \\ \mathrm{Im}[\dot{f}(t)] \end{pmatrix} = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Re}[f(t)] \\ \mathrm{Im}[f(t)] \end{pmatrix}$$

with discrete-time solution:

$$\begin{pmatrix} \mathrm{Re}[f_{k+1}] \\ \mathrm{Im}[f_{k+1}] \end{pmatrix} = \begin{pmatrix} \cos\omega\Delta_k & -\sin\omega\Delta_k \\ \sin\omega\Delta_k & \cos\omega\Delta_k \end{pmatrix} \begin{pmatrix} \mathrm{Re}[f_k] \\ \mathrm{Im}[f_k] \end{pmatrix}$$

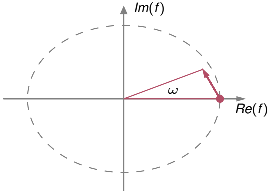
where $\Delta_k = t_{k+1} - t_k$

The cosine kernel

$$K(t, t) = \cos(\omega(t - t'))$$

state space representation is actually a complex ODE:

$$\begin{pmatrix} \mathrm{Re}[\dot{f}(t)] \\ \mathrm{Im}[\dot{f}(t)] \end{pmatrix} = \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix} \begin{pmatrix} \mathrm{Re}[f(t)] \\ \mathrm{Im}[f(t)] \end{pmatrix}$$



The quasi-periodic kernel

So far we have seen the exponential kernel, and the cosine kernel.

We can construct a flexible quasi-periodic kernel, and its SDE form, via their product.

$$K(t, t) = \sigma^2 \exp(-|t - t'|/\ell) \cos(\omega(t - t'))$$

The quasi-periodic kernel

So far we have seen the exponential kernel, and the cosine kernel.

We can construct a flexible quasi-periodic kernel, and its SDE form, via their product.

$$K(t, t) = \sigma^2 \exp(-|t - t'|/\ell) \cos(\omega(t - t'))$$

$$\begin{aligned}
\mathbf{F}_{\cos} &= \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix}, & \mathbf{F}_{\exp} &= -\frac{1}{\ell}, \\
\mathbf{Q}_{c,\cos} &= \text{N/A}, & \mathbf{Q}_{c,\exp} &= \frac{2\sigma^2}{\ell}, \\
\mathbf{L}_{\cos} &= \text{N/A}, & \mathbf{L}_{\exp} &= 1, \\
\mathbf{P}_{\infty,\cos} &= \mathbf{I}_2, & \mathbf{P}_{\infty,\exp} &= \sigma^2, \\
\mathbf{h}_{\cos}^{\top} &= \begin{pmatrix} 1 & 0 \end{pmatrix}, & \mathbf{h}_{\exp}^{\top} &= 1.
\end{aligned}$$

The quasi-periodic kernel

So far we have seen the exponential kernel, and the cosine kernel.

We can construct a flexible quasi-periodic kernel, and its SDE form, via their product.

$$K(t, t) = \sigma^2 \exp(-|t - t'|/\ell) \cos(\omega(t - t'))$$

$$\frac{\mathrm{d}\mathbf{f}(t)}{\mathrm{d}t} = \mathbf{F}\mathbf{f}(t) + \mathbf{L}\mathbf{w}(t)$$

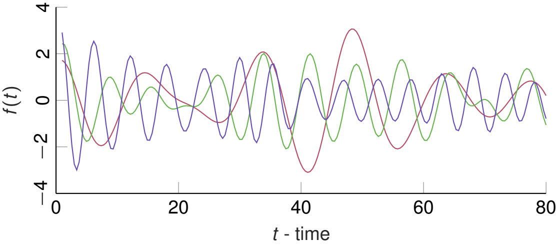
$$\begin{aligned}
\mathbf{F} &= \mathbf{F}_{\cos} \oplus \mathbf{F}_{\exp} &&= \mathbf{F}_{\cos} \otimes \mathbf{I}_2 + \mathbf{I}_1 \otimes \mathbf{F}_{\exp} = \begin{pmatrix} -\frac{1}{\ell} & -\omega \\ \omega & -\frac{1}{\ell} \end{pmatrix}, \\
\mathbf{Q}_c &= \mathbf{I}_2 \otimes \mathbf{Q}_{c,\exp} &&= \frac{2\sigma^2}{\ell} \mathbf{I}_2, \\
\mathbf{L} &= \mathbf{I}_2 \otimes \mathbf{L}_{\exp} &&= \mathbf{I}_2, \\
\mathbf{P}_\infty &= \mathbf{I}_2 \otimes \mathbf{P}_{\infty,\exp} &&= \sigma^2 \mathbf{I}_2.
\end{aligned}$$

The quasi-periodic kernel

So far we have seen the exponential kernel, and the cosine kernel.

We can construct a flexible quasi-periodic kernel, and its SDE form, via their product.

$$K(t,t) = \sigma^2 \exp(-|t-t'|/\ell)\cos(\omega(t-t'))$$



Sample paths from the quasi-periodic GP

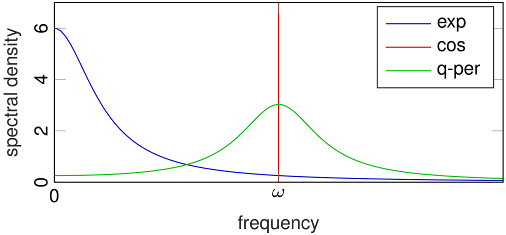
The quasi-periodic kernel

So far we have seen the exponential kernel, and the cosine kernel.

We can construct a flexible quasi-periodic kernel, and its SDE form, via their product.

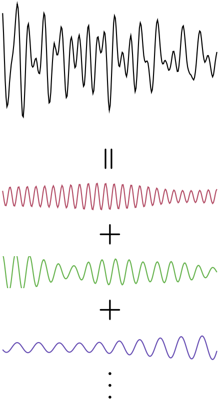
$$K(t, t) = \sigma^2 \exp(-|t - t'|/\ell) \cos(\omega(t - t'))$$

Constructing the quasi-periodic kernel in the spectral domain



The cosine kernel acts as a frequency shift operator on the exponential kernel to produce the quasi-periodic kernel ($\sigma^2 = 1$, $\ell = 3$, $\omega = \pi/2$).

Spectral mixtures



Spectral mixtures

$$f(t) \sim \mathcal{GP}\left(0, \sum_{d=1}^{D} K_{\mathrm{q-per}}^{(d)}(t, t')\right), \qquad \mathbf{f}_k = \mathbf{A}_k \mathbf{f}_{k-1} + \mathbf{q}_{k-1},$$

$$y_k \sim \mathrm{N}(f(t_k), \sigma_y^2) \qquad y_k = \mathbf{h}^\top \mathbf{f}_k + \sigma_y \epsilon_{y,k}$$

Spectral mixtures

$$f(t) \sim \mathcal{GP}\left(0, \sum_{d=1}^{D} K^{(d)}_{\text{q-per}}(t,t')\right), \qquad \mathbf{f}_k = \mathbf{A}_k \mathbf{f}_{k-1} + \mathbf{q}_{k-1},$$

$$y_k \sim \mathrm{N}(f(t_k), \sigma_y^2) \qquad y_k = \mathbf{h}^\top \mathbf{f}_k + \sigma_y \epsilon_{y,k}$$

$$\mathbf{A}_k = \begin{pmatrix} \exp\left(-\frac{\Delta_k}{\ell_1}\right)\begin{pmatrix} \cos\omega_1\Delta_k & -\sin\omega_1\Delta_k \\ \sin\omega_1\Delta_k & \cos\omega_1\Delta_k \end{pmatrix} & & \\ & \ddots & \\ & & \exp\left(-\frac{\Delta_k}{\ell_D}\right)\begin{pmatrix} \cos\omega_D\Delta_k & -\sin\omega_D\Delta_k \\ \sin\omega_D\Delta_k & \cos\omega_D\Delta_k \end{pmatrix} \end{pmatrix},$$

$$\mathbf{Q}_k = \begin{pmatrix} \sigma_1^2(1-\exp(-\frac{2\Delta_k}{\ell_1}))\mathbf{I}_2 & & \\ & \ddots & \\ & & \sigma_D^2(1-\exp(-\frac{2\Delta_k}{\ell_D}))\mathbf{I}_2 \end{pmatrix}.$$

$$\mathbf{h}^\top = \begin{pmatrix} 1 & 0 & 1 & 0 & \ldots & 1 & 0 \end{pmatrix}.$$

Spectral mixtures

$$f(t) \sim \mathcal{GP}\left(0, \sum_{d=1}^{D} K^{(d)}_{\mathrm{q-per}}(t,t')\right), \qquad \mathbf{f}_k = \mathbf{A}_k \mathbf{f}_{k-1} + \mathbf{q}_{k-1},$$

$$y_k \sim \mathrm{N}(f(t_k), \sigma_y^2) \qquad y_k = \mathbf{h}^\top \mathbf{f}_k + \sigma_y \epsilon_{y,k}$$

Can equivalently be written as:

$$f_{d,k} = \psi_d \mathrm{e}^{i\omega_d \Delta_k} f_{d,k-1} + \rho_d \, \epsilon_{d,k},$$

$$y_k = \sum_{d=1}^{D} \mathrm{Re}[f_{d,k}] + \sigma_y \, \epsilon_{y,k},$$

with $\psi_d = \exp(-\Delta t/\ell_d)$ and $\rho_d = \sigma_d^2(1 - \exp(-2\Delta t/\ell_d))$.

Spectral mixtures

$$f(t) \sim \mathcal{GP}\left(0, \sum_{d=1}^{D} K_{\mathrm{q-per}}^{(d)}(t, t')\right), \qquad \mathbf{f}_k = \mathbf{A}_k \mathbf{f}_{k-1} + \mathbf{q}_{k-1},$$

$$y_k \sim \mathrm{N}(f(t_k), \sigma_y^2) \qquad y_k = \mathbf{h}^\top \mathbf{f}_k + \sigma_y \epsilon_{y,k}$$

Can equivalently be written as:

$$f_{d,k} = \psi_d \mathrm{e}^{i\omega_d \Delta_k} f_{d,k-1} + \rho_d \, \epsilon_{d,k},$$

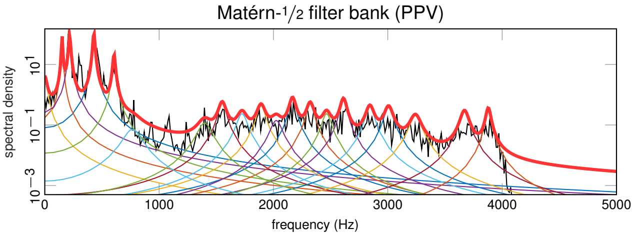
$$y_k = \sum_{d=1}^{D} \mathrm{Re}[f_{d,k}] + \sigma_y \, \epsilon_{y,k},$$

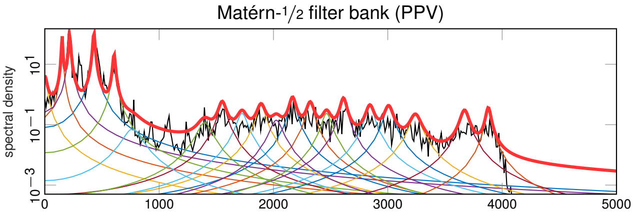
with $\psi_d = \exp(-\Delta t/\ell_d)$ and $\rho_d = \sigma_d^2(1 - \exp(-2\Delta t/\ell_d))$.

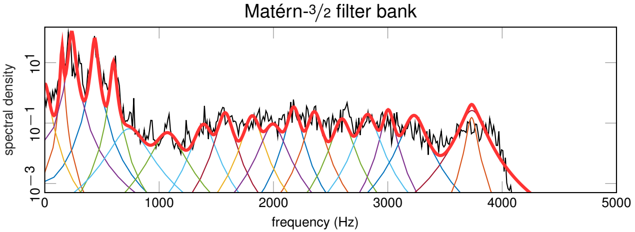
This model is known as the probabilistic phase vocoder.

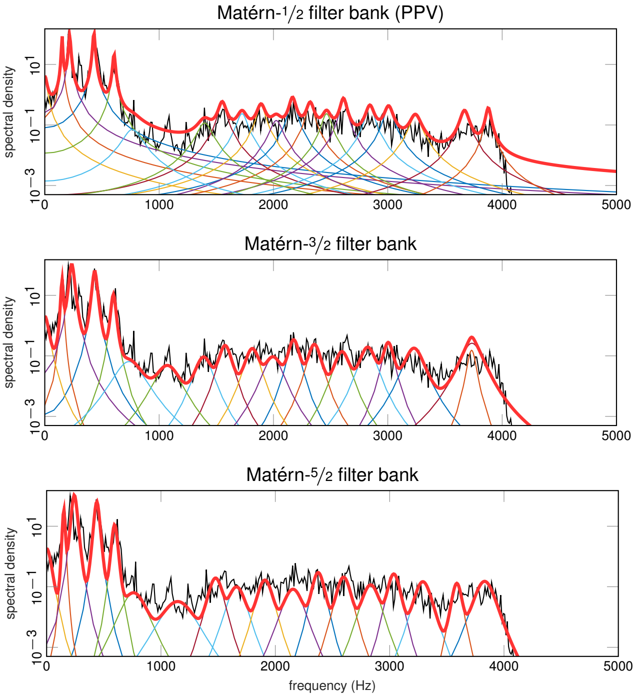
Comparing the two model interpretations

Probabilistic phase vocoder	Spectral mixture GP
Fast inference via Kalman smoothing	Inference slow for long time-series
Fast frequency-domain parameter learning	Freq.-domain parameter learning possible
Interpreting the model can be challenging	All model assumptions encoded in the kernel
Changing the model is hard	Changing the model is easy


Matérn-1/2 filter bank (PPV)
spectral density
10^{1}
10^{-1}
10^{-3}
0
1000
2000
3000
4000
5000
frequency (Hz)

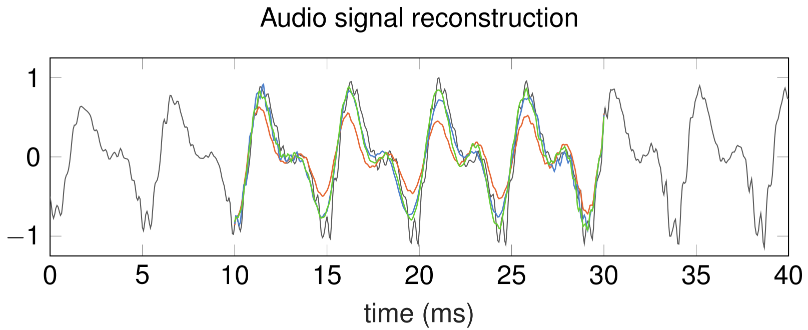

Matérn-1/2 filter bank (PPV)
spectral density
10^{-3}
10^{-1}
10^{1}
0
1000
2000
3000
4000
5000


Matérn-3/2 filter bank
spectral density
10^{-3}
10^{-1}
10^{1}
0
1000
2000
3000
4000
5000
frequency (Hz)



Matérn-1/2 filter bank (PPV)
spectral density
10^{-3}
10^{-1}
10^{1}
0
1000
2000
3000
4000
5000
Matérn-3/2 filter bank
spectral density
10^{-3}
10^{-1}
10^{1}
0
1000
2000
3000
4000
5000
Matérn-5/2 filter bank
spectral density
10^{-3}
10^{-1}
10^{1}
0
1000
2000
3000
4000
5000
frequency (Hz)

Missing Data Synthesis



Data imputation using a filter bank composed of the following kernels:
Matérn$^1/_2$ (exponential) - 1^{st} *order state space form*
Matérn$^3/_2$ - 2^{nd} *order state space form*
Matérn$^5/_2$ - 3^{rd} *order state space form*

PART II - APPROXIMATE INFERENCE IN TEMPORAL GPs

Non-conjugate state space models

$$
\begin{aligned}
f(t) &\sim \mathcal{GP}\big(0,\, K_\theta(t,t')\big), \\
y_k &\sim p(y_k \mid f(t_k))
\end{aligned}
\qquad\qquad
\begin{aligned}
\mathbf{f}_k &= \mathbf{A}_{\theta,k}\mathbf{f}_{k-1} + \mathbf{q}_{k-1}, \\
y_k &\sim p(y_k \mid \mathbf{h}^\top \mathbf{f}_k)
\end{aligned}
$$

Non-conjugate state space models

$$
\begin{aligned}
f(t) &\sim \mathcal{GP}\big(0,\, K_\theta(t,t')\big), \\
y_k &\sim p(y_k \mid f(t_k))
\end{aligned}
\qquad\qquad
\begin{aligned}
\mathbf{f}_k &= \mathbf{A}_{\theta,k}\mathbf{f}_{k-1} + \mathbf{q}_{k-1}, \\
y_k &\sim p(y_k \mid \mathbf{h}^\top \mathbf{f}_k)
\end{aligned}
$$

Many approximate inference methods (VB, EP) can be applied in the state space regime.

Expectation propagation (EP)

Intuitively a good fit for time series — we can process the data sequentially.

Expectation propagation (EP)

A closer look at the Kalman filter:

Expectation propagation (EP)

A closer look at the Kalman filter:

predict step:

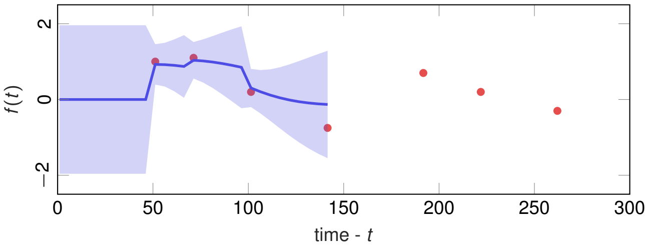
$$
\begin{aligned}
p(\mathbf{f}_k | y_{1:k-1}) = \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-) &= \int p(\mathbf{f}_k, \mathbf{f}_{k-1} \mid y_{1:k-1}) \, \mathrm{d}\mathbf{f}_{k-1} \\
&= \int p(\mathbf{f}_k \mid \mathbf{f}_{k-1}) p(\mathbf{f}_{k-1} \mid y_{1:k-1}) \, \mathrm{d}\mathbf{f}_{k-1} \\
&= \int \mathrm{N}(\mathbf{A}_k, \mathbf{Q}_k) \mathrm{N}(\mathbf{m}_{k-1}, \mathbf{P}_{k-1}) \, \mathrm{d}\mathbf{f}_{k-1} \\
&= \mathrm{N}(\mathbf{A}_k \mathbf{m}_{k-1},\ \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^\top + \mathbf{Q}_k)
\end{aligned}
$$

Expectation propagation (EP)

A closer look at the Kalman filter:

predict step:

$$
\begin{aligned}
p(\mathbf{f}_k|y_{1:k-1}) = \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-) &= \int p(\mathbf{f}_k, \mathbf{f}_{k-1} \mid y_{1:k-1})\, \mathrm{d}\mathbf{f}_{k-1} \\
&= \int p(\mathbf{f}_k \mid \mathbf{f}_{k-1}) p(\mathbf{f}_{k-1} \mid y_{1:k-1})\, \mathrm{d}\mathbf{f}_{k-1} \\
&= \int \mathrm{N}(\mathbf{A}_k, \mathbf{Q}_k) \mathrm{N}(\mathbf{m}_{k-1}, \mathbf{P}_{k-1})\, \mathrm{d}\mathbf{f}_{k-1} \\
&= \mathrm{N}(\mathbf{A}_k \mathbf{m}_{k-1},\ \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^\top + \mathbf{Q}_k)
\end{aligned}
$$

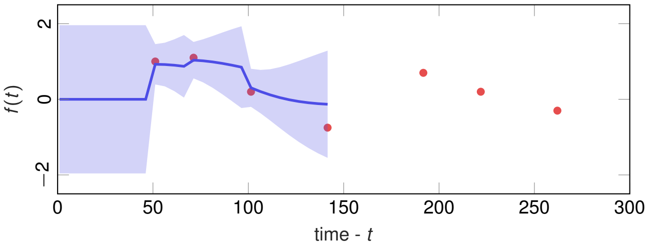


Expectation propagation (EP)

A closer look at the Kalman filter:

update step:

$$
\begin{aligned}
p(\mathbf{f}_k|y_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k) &\propto p(\mathbf{f}_k|y_{1:k-1})p(y_k \mid f(t_k)) \\
&= \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)\, p(y_k \mid f(t_k))
\end{aligned}
$$



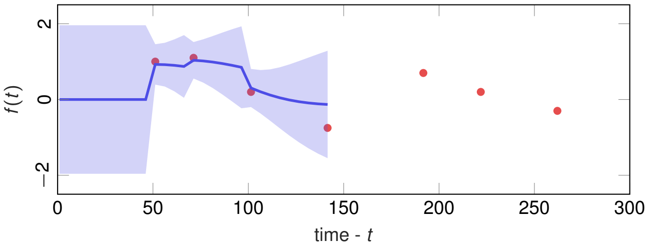
Expectation propagation (EP)

A closer look at the Kalman filter:

update step:

$$p(\mathbf{f}_k | y_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k) \propto p(\mathbf{f}_k | y_{1:k-1}) p(y_k \mid f(t_k))$$
$$= \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)\, p(y_k \mid f(t_k))$$

if $p(y_k \mid f(t_k)) \sim \mathrm{N}(\cdot, \cdot)$, the Kalman update equations are just a stable way to calculate this product of Gaussian densities.



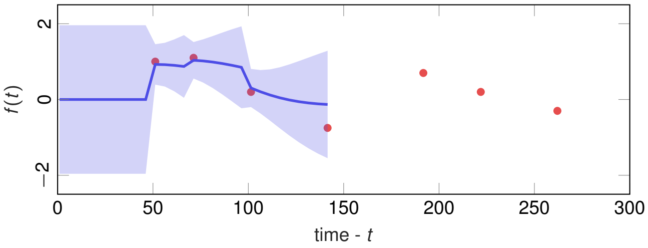
Expectation propagation (EP)

A closer look at the Kalman filter:

update step:

$$
\begin{aligned}
p(\mathbf{f}_k|y_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k) &\propto p(\mathbf{f}_k|y_{1:k-1})p(y_k \mid f(t_k)) \\
&= \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)\, p(y_k \mid f(t_k))
\end{aligned}
$$

How about if $p(y_k \mid f(t_k))$ is not Gaussian?

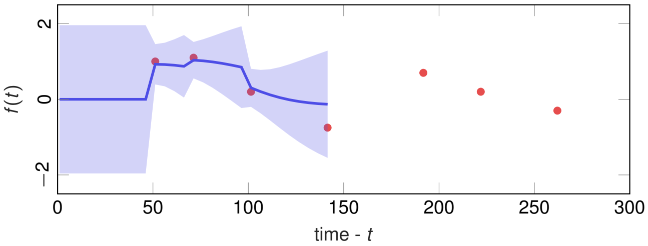


Expectation propagation (EP)

A closer look at the Kalman filter:

update step:

$$p(\mathbf{f}_k|y_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k) \propto p(\mathbf{f}_k|y_{1:k-1})p(y_k \mid f(t_k))$$
$$= \underbrace{\mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)}_{\substack{\text{“prior” over } \mathbf{f}_k \\ \text{conditioned on} \\ \text{past data}}} p(y_k \mid f(t_k))$$

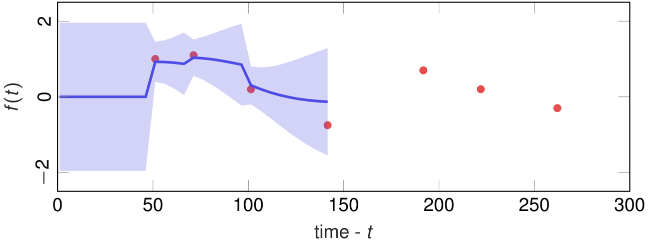


Expectation propagation (EP)

A closer look at the Kalman filter:

update step:

$$
\begin{aligned}
p(\mathbf{f}_k|y_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k) &\propto p(\mathbf{f}_k|y_{1:k-1})p(y_k \mid f(t_k)) \\
&= \underbrace{\mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)}_{\text{“cavity distribution”}} p(y_k \mid f(t_k))
\end{aligned}
$$

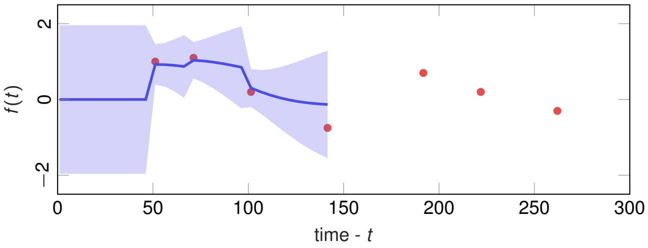


Expectation propagation (EP)

A closer look at the Kalman filter:

update step:

$$
\begin{aligned}
p(\mathbf{f}_k | y_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k) &\propto p(\mathbf{f}_k | y_{1:k-1}) p(y_k \mid f(t_k)) \\
&= \underbrace{\mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)\, p(y_k \mid f(t_k))}_{\text{“tilted distribution”}}
\end{aligned}
$$

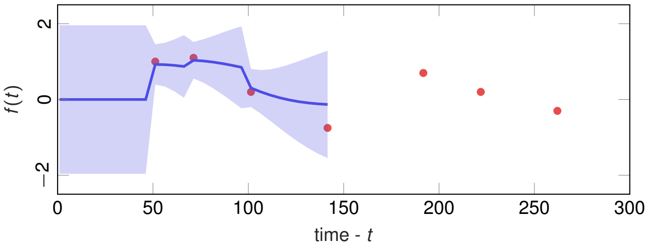


Expectation propagation (EP)

A closer look at the Kalman filter:

update step:

$$
\begin{aligned}
p(\mathbf{f}_k | y_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k) &\propto p(\mathbf{f}_k | y_{1:k-1}) p(y_k \mid f(t_k)) \\
&= \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)\, p(y_k \mid f(t_k)) \\
&\approx \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)\, \underbrace{s(\mathbf{f}_k)}_{\text{“site”}}
\end{aligned}
$$



Expectation propagation (EP)

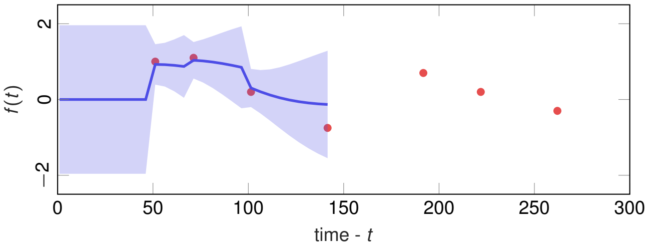
A closer look at the Kalman filter:

update step:

$$
\begin{aligned}
p(\mathbf{f}_k | y_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k) &\propto p(\mathbf{f}_k | y_{1:k-1}) p(y_k \mid f(t_k)) \\
&= \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)\, p(y_k \mid f(t_k)) \\
&\approx \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)\, s(\mathbf{f}_k)
\end{aligned}
$$

EP update:

match moments



Expectation propagation (EP)

A closer look at the Kalman filter:

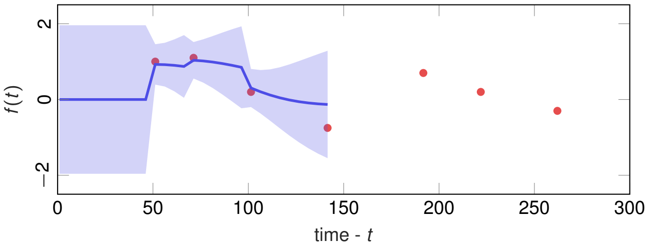
update step:

$$
\begin{aligned}
p(\mathbf{f}_k | y_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k) &\propto p(\mathbf{f}_k | y_{1:k-1}) p(y_k \mid f(t_k)) \\
&= \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)\, p(y_k \mid f(t_k)) \\
&\approx \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)\, s(\mathbf{f}_k)
\end{aligned}
$$

EP update:

match moments

i.e., choose $s(\mathbf{f}_k) \sim \mathrm{N}(\mathbf{m}_k^{\text{site}}, \mathbf{P}_k^{\text{site}})$ such that the moments are matched. Store to be refined later.



Expectation propagation (EP)

A closer look at the Kalman filter:

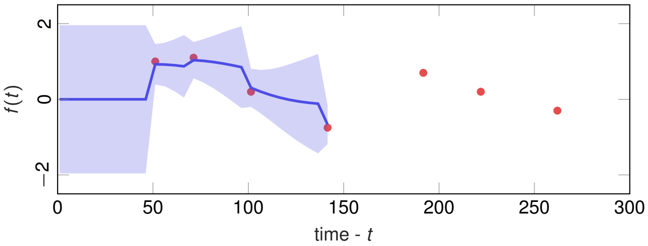
update step:

$$
\begin{aligned}
p(\mathbf{f}_k | y_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k) &\propto p(\mathbf{f}_k | y_{1:k-1}) p(y_k \mid f(t_k)) \\
&= \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)\, p(y_k \mid f(t_k)) \\
&\approx \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)\, s(\mathbf{f}_k)
\end{aligned}
$$

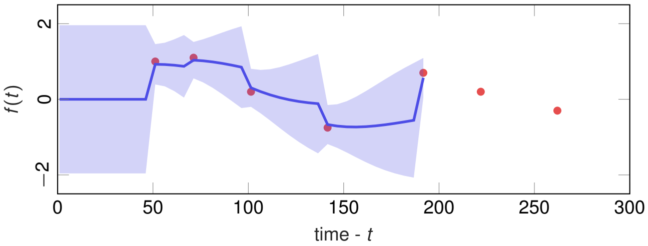
EP update:

match moments

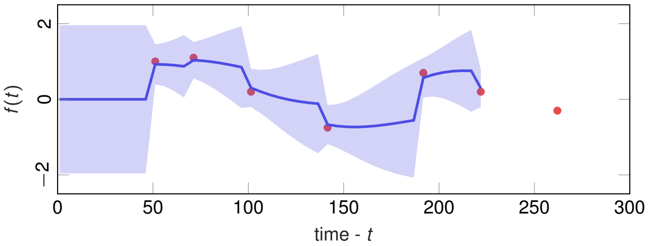
i.e., choose $s(\mathbf{f}_k) \sim \mathrm{N}(\mathbf{m}_k^{\text{site}}, \mathbf{P}_k^{\text{site}})$ such that the moments are matched. Store to be refined later.



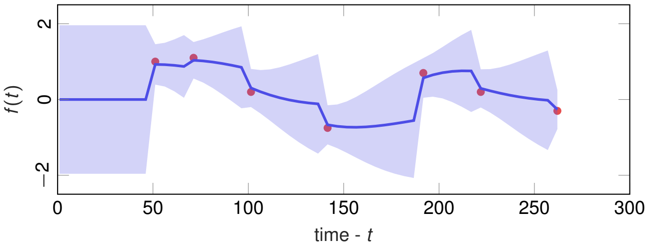
Expectation propagation (EP)



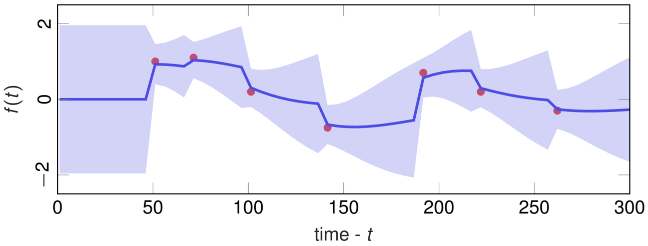
Expectation propagation (EP)



Expectation propagation (EP)



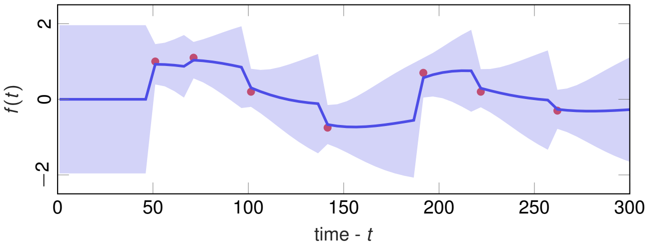
Expectation propagation (EP)



Expectation propagation (EP)

Now consider the RTS Smoother:

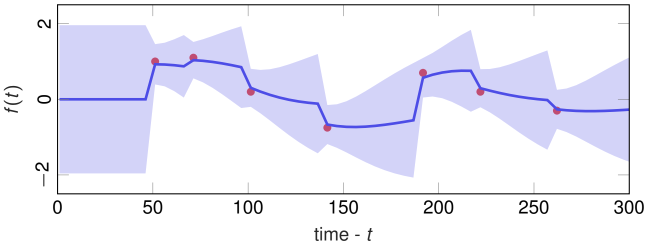
- update predictions with future observations
- refine the EP sites along the way



Expectation propagation (EP)

Now consider the RTS Smoother:

$$p(\mathbf{f}_k | y_{1:T}) \propto p(\mathbf{f}_k \mid \mathbf{y}_{k+1:N}) p(\mathbf{f}_k \mid \mathbf{y}_{1:k})$$

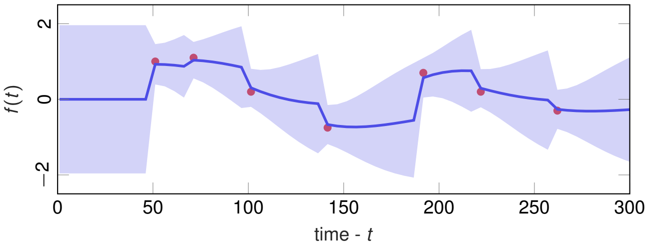


Expectation propagation (EP)

Now consider the RTS Smoother:

We have the full (marginal) posterior, so we must explicitly remove the sites:

$$p_{\text{cavity}}(\mathbf{f}_k) = p(\mathbf{f}_k | y_{1:T}) / s_{\text{old}}(\mathbf{f}_k)$$



Expectation propagation (EP)

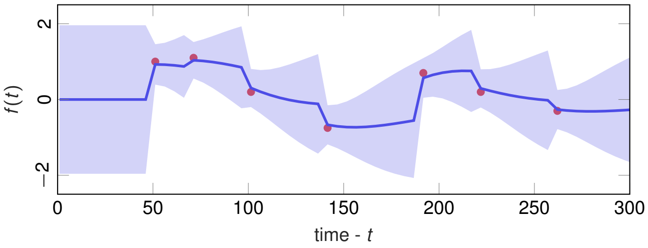
Now consider the RTS Smoother:

We have the full (marginal) posterior, so we must explicitly remove the sites:

$$p_{\text{cavity}}(\mathbf{f}_k) = p(\mathbf{f}_k|y_{1:T})/s_{\text{old}}(\mathbf{f}_k)$$

Tilted distribution:

$$\begin{aligned} p(\mathbf{f}_k|y_{1:T}) &= p_{\text{cavity}}(\mathbf{f}_k)p(y_k \mid f(t_k)) \\ &\approx p_{\text{cavity}}(\mathbf{f}_k)s_{\text{new}}(\mathbf{f}_k) \end{aligned}$$



Expectation propagation (EP)

Now consider the RTS Smoother:

We have the full (marginal) posterior, so we must explicitly remove the sites:

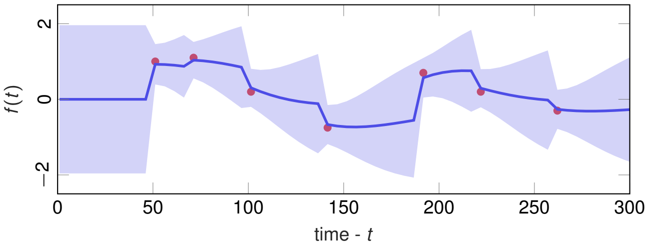
$$p_{\text{cavity}}(\mathbf{f}_k) = p(\mathbf{f}_k | y_{1:T}) / s_{\text{old}}(\mathbf{f}_k)$$

Tilted distribution:

$$\begin{aligned} p(\mathbf{f}_k | y_{1:T}) &= p_{\text{cavity}}(\mathbf{f}_k) p(y_k \mid f(t_k)) \\ &\approx p_{\text{cavity}}(\mathbf{f}_k) s_{\text{new}}(\mathbf{f}_k) \end{aligned}$$

EP update:

match moments



Expectation propagation (EP)

Now consider the RTS Smoother:

We have the full (marginal) posterior, so we must explicitly remove the sites:

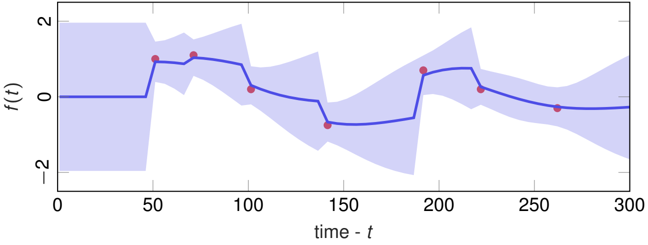
$$p_{\text{cavity}}(\mathbf{f}_k) = p(\mathbf{f}_k | y_{1:T}) / s_{\text{old}}(\mathbf{f}_k)$$

Tilted distribution:

$$\begin{aligned} p(\mathbf{f}_k | y_{1:T}) &= p_{\text{cavity}}(\mathbf{f}_k) p(y_k \mid f(t_k)) \\ &\approx p_{\text{cavity}}(\mathbf{f}_k) s_{\text{new}}(\mathbf{f}_k) \end{aligned}$$

EP update:

match moments



Expectation propagation (EP)

Now consider the RTS Smoother:

We have the full (marginal) posterior, so we must explicitly remove the sites:

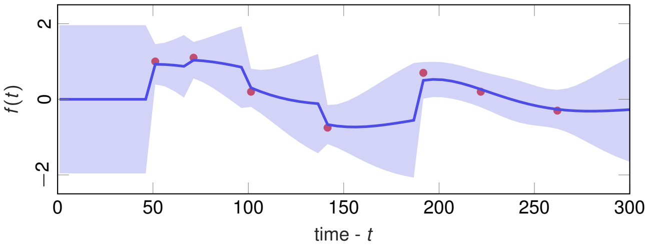
$$p_{\text{cavity}}(\mathbf{f}_k) = p(\mathbf{f}_k | y_{1:T}) / s_{\text{old}}(\mathbf{f}_k)$$

Tilted distribution:

$$\begin{aligned} p(\mathbf{f}_k | y_{1:T}) &= p_{\text{cavity}}(\mathbf{f}_k) p(y_k \mid f(t_k)) \\ &\approx p_{\text{cavity}}(\mathbf{f}_k) s_{\text{new}}(\mathbf{f}_k) \end{aligned}$$

EP update:

match moments



Expectation propagation (EP)

Now consider the RTS Smoother:

We have the full (marginal) posterior, so we must explicitly remove the sites:

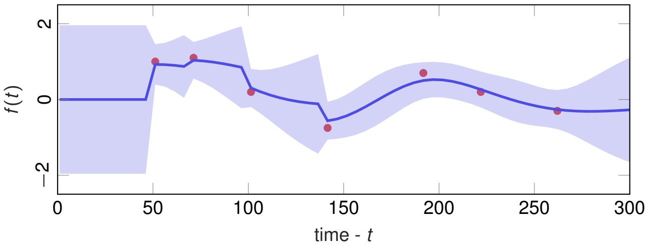
$$p_{\text{cavity}}(\mathbf{f}_k) = p(\mathbf{f}_k | y_{1:T}) / s_{\text{old}}(\mathbf{f}_k)$$

Tilted distribution:

$$\begin{aligned} p(\mathbf{f}_k | y_{1:T}) &= p_{\text{cavity}}(\mathbf{f}_k) p(y_k \mid f(t_k)) \\ &\approx p_{\text{cavity}}(\mathbf{f}_k) s_{\text{new}}(\mathbf{f}_k) \end{aligned}$$

EP update:

match moments



Expectation propagation (EP)

Now consider the RTS Smoother:

We have the full (marginal) posterior, so we must explicitly remove the sites:

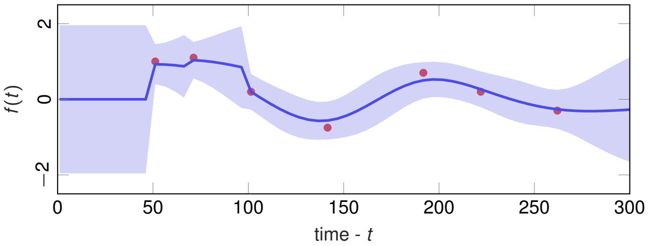
$$p_{\text{cavity}}(\mathbf{f}_k) = p(\mathbf{f}_k | y_{1:T}) / s_{\text{old}}(\mathbf{f}_k)$$

Tilted distribution:

$$\begin{aligned} p(\mathbf{f}_k | y_{1:T}) &= p_{\text{cavity}}(\mathbf{f}_k) p(y_k \mid f(t_k)) \\ &\approx p_{\text{cavity}}(\mathbf{f}_k) s_{\text{new}}(\mathbf{f}_k) \end{aligned}$$

EP update:

match moments



Expectation propagation (EP)

Now consider the RTS Smoother:

We have the full (marginal) posterior, so we must explicitly remove the sites:

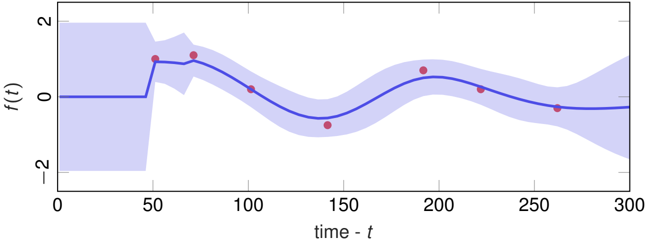
$$p_{\text{cavity}}(\mathbf{f}_k) = p(\mathbf{f}_k | y_{1:T}) / s_{\text{old}}(\mathbf{f}_k)$$

Tilted distribution:

$$\begin{aligned} p(\mathbf{f}_k | y_{1:T}) &= p_{\text{cavity}}(\mathbf{f}_k) p(y_k \mid f(t_k)) \\ &\approx p_{\text{cavity}}(\mathbf{f}_k) s_{\text{new}}(\mathbf{f}_k) \end{aligned}$$

EP update:

match moments



Expectation propagation (EP)

Now consider the RTS Smoother:

We have the full (marginal) posterior, so we must explicitly remove the sites:

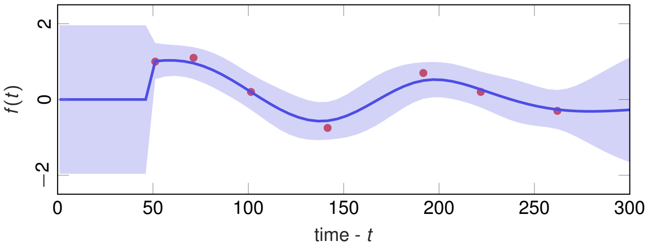
$$p_{\text{cavity}}(\mathbf{f}_k) = p(\mathbf{f}_k | y_{1:T}) / s_{\text{old}}(\mathbf{f}_k)$$

Tilted distribution:

$$\begin{aligned} p(\mathbf{f}_k | y_{1:T}) &= p_{\text{cavity}}(\mathbf{f}_k) p(y_k \mid f(t_k)) \\ &\approx p_{\text{cavity}}(\mathbf{f}_k) s_{\text{new}}(\mathbf{f}_k) \end{aligned}$$

EP update:

match moments



Expectation propagation (EP)

Now consider the RTS Smoother:

We have the full (marginal) posterior, so we must explicitly remove the sites:

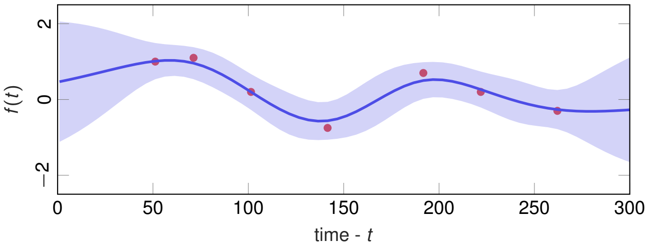
$$p_{\text{cavity}}(\mathbf{f}_k) = p(\mathbf{f}_k | y_{1:T}) / s_{\text{old}}(\mathbf{f}_k)$$

Tilted distribution:

$$\begin{aligned} p(\mathbf{f}_k | y_{1:T}) &= p_{\text{cavity}}(\mathbf{f}_k) p(y_k \mid f(t_k)) \\ &\approx p_{\text{cavity}}(\mathbf{f}_k) s_{\text{new}}(\mathbf{f}_k) \end{aligned}$$

EP update:

match moments



Expectation propagation (EP)

Now consider the RTS Smoother:

We have the full (marginal) posterior, so we must explicitly remove the sites:

$$p_{\text{cavity}}(\mathbf{f}_k) = p(\mathbf{f}_k | y_{1:T}) / s_{\text{old}}(\mathbf{f}_k)$$

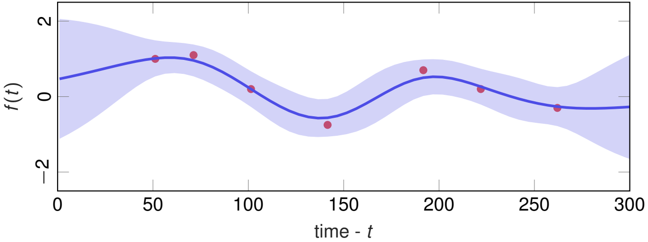
Tilted distribution:

$$\begin{aligned} p(\mathbf{f}_k | y_{1:T}) &= p_{\text{cavity}}(\mathbf{f}_k) p(y_k \mid f(t_k)) \\ &\approx p_{\text{cavity}}(\mathbf{f}_k) s_{\text{new}}(\mathbf{f}_k) \end{aligned}$$

EP update:

match moments

The new sites $s_{\text{new}}(\mathbf{f}_k)$ will be used on the next forward pass.



Expectation propagation (EP)

Now consider the RTS Smoother:

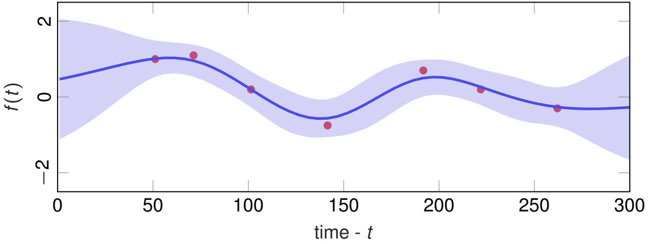
We have the full (marginal) posterior, so we must explicitly remove the sites:

$$p_{\text{cavity}}(\mathbf{f}_k) = p(\mathbf{f}_k | y_{1:T}) / s^{\alpha}_{\text{old}}(\mathbf{f}_k)$$

Tilted distribution:

$$\begin{aligned} p(\mathbf{f}_k | y_{1:T}) &= p_{\text{cavity}}(\mathbf{f}_k) p^{\alpha}(y_k \mid f(t_k)) \\ &\approx p_{\text{cavity}}(\mathbf{f}_k) s^{\alpha}_{\text{new}}(\mathbf{f}_k) \end{aligned}$$

Can add in the usual EP extras: power (α) and damping



An example: nonstationary TF-analysis

We apply this power EP method to a nonstationary extension of time-frequency analysis model:

An example: nonstationary TF-analysis

We apply this power EP method to a nonstationary extension of time-frequency analysis model:

prior:

$$f_d(t) \sim \mathrm{GP}(0, K^{(d)}_{\mathrm{q-periodic}}(t,t')), \qquad d = 1, 2, \ldots, D$$
$$\log a_d(t) \sim \mathrm{GP}(0, K^{(n)}_{\mathrm{Matérn}}(t,t')),$$

An example: nonstationary TF-analysis

We apply this power EP method to a nonstationary extension of time-frequency analysis model:

prior:

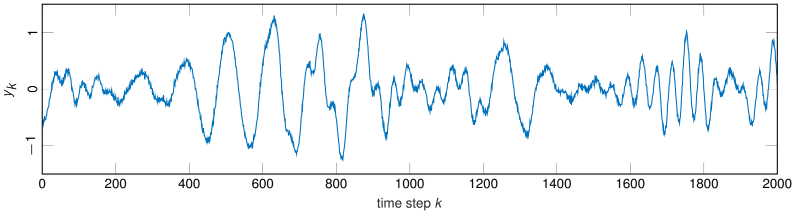
$$f_d(t) \sim \mathrm{GP}(0, K^{(d)}_{\mathrm{q-periodic}}(t,t')), \qquad d = 1, 2, \ldots, D$$
$$\log a_d(t) \sim \mathrm{GP}(0, K^{(n)}_{\mathrm{Matérn}}(t,t')),$$

likelihood:

$$y_k = \sum_d a_d(t_k)\, f_d(t_k) + \sigma_y\, \epsilon_k$$

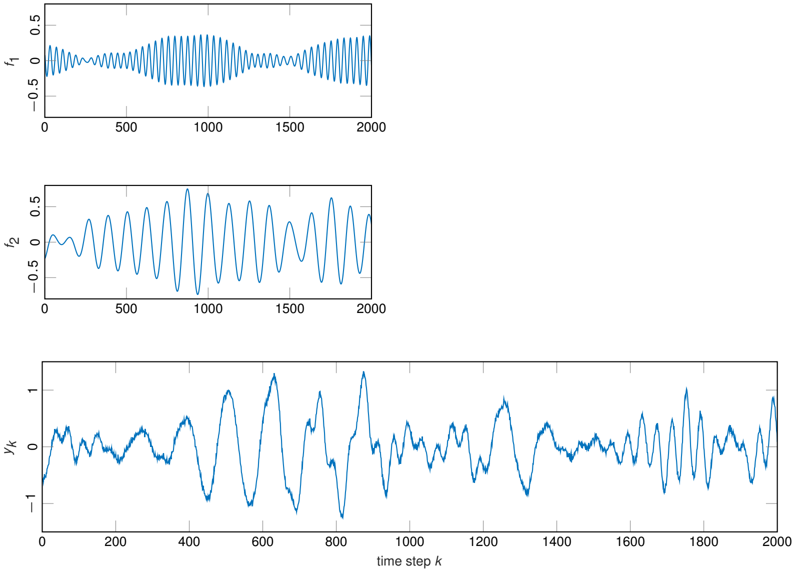
- $f_d(t)$: frequency components
- $a_d(t)$: positive amplitudes

Toy data



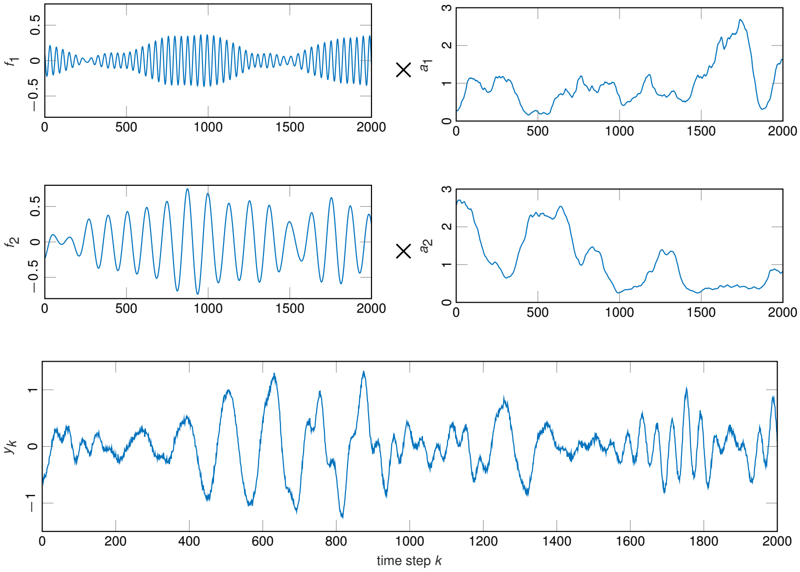
y_k
1
0
−1
0
200
400
600
800
1000
1200
1400
1600
1800
2000
time step k

Toy data

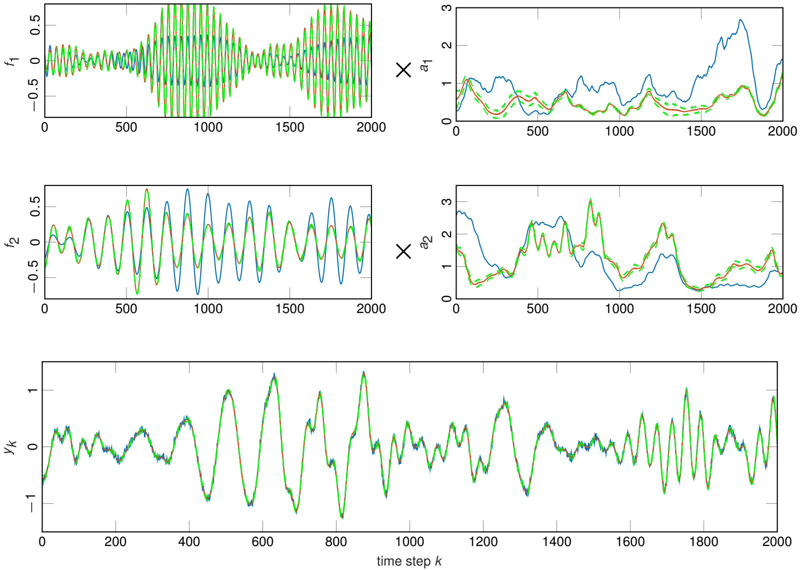


f_1
-0.5
0
0.5
0
500
1000
1500
2000
f_2
-0.5
0
0.5
0
500
1000
1500
2000
y_k
-1
0
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000
time step k

Toy data



Toy data



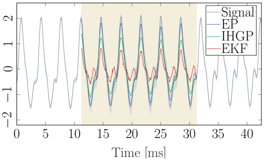
Applications

The model can, without modification, be applied to:

Applications

The model can, without modification, be applied to:

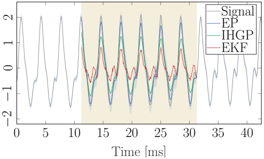
Missing Data Synthesis



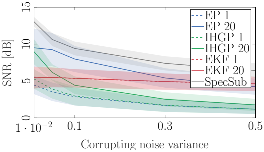
Applications

The model can, without modification, be applied to:

Missing Data Synthesis



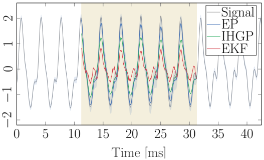
Denoising



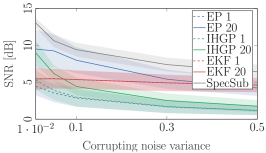
Applications

The model can, without modification, be applied to:

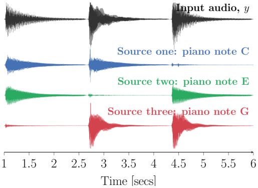
Missing Data Synthesis



Denoising



Source Separation



One issue with the power EP method

The crucial moment matching step involves calculating the following (intractable) expectation:

$$\mathcal{Z}_{\text{tilted}} = \mathbb{E}_{p_{\text{cavity}}(\mathbf{f}_k)}\left[p(y_k \mid \mathbf{f}_k)^{\alpha}\right]$$

One issue with the power EP method

The crucial moment matching step involves calculating the following (intractable) expectation:

$$\mathcal{Z}_{\text{tilted}} = \mathbb{E}_{p_{\text{cavity}}(\mathbf{f}_k)} \left[p(y_k \mid \mathbf{f}_k)^{\alpha} \right]$$

This can be a very high-dimensional integral for some models!

PART III - SCALABLE GLOBAL INFERENCE VIA LOCAL LINEARISATION

Another contradiction

Approximate inference in practice

In practical (industrial) applications, the extended Kalman filter (EKF) is still the tool of choice

Modern day approximate inference (VB, EP, ...) is more general and better approximates the true posterior

Pseudo-code for Kalman filter EP

for $k = 1 : T$

Kalman predict:

$$p(\mathbf{f}_k|\mathbf{y}_{1:k-1}) = \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-) = \mathrm{N}(\mathbf{A}_k\mathbf{m}_{k-1},\ \mathbf{A}_k\mathbf{P}_{k-1}\mathbf{A}_k^\top + \mathbf{Q}_k)$$

Moment match:

$$\mathcal{L}_k = \log \mathbb{E}_{\mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)}\left[p(\mathbf{y}_k \mid \mathbf{f}_k)^\alpha\right],$$

$$\mathbf{m}_k^{\text{site}} = \mathbf{m}_k^- - \left(\frac{\mathrm{d}^2\mathcal{L}_k}{\mathrm{d}{\mathbf{m}_k}^2}\right)^{-1}\frac{\mathrm{d}\mathcal{L}_k}{\mathrm{d}\mathbf{m}_k}, \quad \mathbf{P}_k^{\text{site}} = \alpha\left(-\mathbf{P}_k^- - \left(\frac{\mathrm{d}^2\mathcal{L}_k}{\mathrm{d}{\mathbf{m}_k}^2}\right)^{-1}\right).$$

Kalman update:

$$\mathbf{S}_k = \mathbf{P}_k^- + \mathbf{P}_k^{\text{site}}, \qquad \mathbf{K}_k = \mathbf{P}_k^-\mathbf{S}_k^{-1},$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k(\mathbf{m}_k^{\text{site}} - \mathbf{m}_k^-), \qquad \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top.$$

$$p(\mathbf{f}_k|\mathbf{y}_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k)$$

end for

Pseudo-code for Kalman filter EP

for $k = 1 : T$

Kalman predict:

$$p(\mathbf{f}_k|\mathbf{y}_{1:k-1}) = \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-) = \mathrm{N}(\mathbf{A}_k \mathbf{m}_{k-1},\, \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^\top + \mathbf{Q}_k)$$

Moment match:

$$\mathcal{L}_k = \log \mathbb{E}_{\mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)} \left[p(\mathbf{y}_k \mid \mathbf{f}_k)^\alpha \right],$$

$$\mathbf{m}_k^{\text{site}} = \mathbf{m}_k^- - \left(\frac{\mathrm{d}^2 \mathcal{L}_k}{\mathrm{d}\mathbf{m}_k{}^2} \right)^{-1} \frac{\mathrm{d}\mathcal{L}_k}{\mathrm{d}\mathbf{m}_k}, \quad \mathbf{P}_k^{\text{site}} = \alpha \left(-\mathbf{P}_k^- - \left(\frac{\mathrm{d}^2 \mathcal{L}_k}{\mathrm{d}\mathbf{m}_k{}^2} \right)^{-1} \right).$$

Kalman update:

$$\mathbf{S}_k = \mathbf{P}_k^- + \mathbf{P}_k^{\text{site}}, \qquad \mathbf{K}_k = \mathbf{P}_k^- \mathbf{S}_k^{-1},$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k (\mathbf{m}_k^{\text{site}} - \mathbf{m}_k^-), \qquad \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^\top.$$

$$p(\mathbf{f}_k|\mathbf{y}_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k)$$

end for

Pseudo-code for Kalman filter EP

$$\mathcal{L}_k = \log \mathbb{E}_{\mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)}\left[p(\mathbf{y}_k \mid \mathbf{f}_k)^\alpha\right]$$

Pseudo-code for Kalman filter EP

Likelihood $p(\mathbf{y}_k \mid \mathbf{f}_k) = h(\mathbf{f}_k, \mathbf{r}_k)$ is a nonlinear function of Gaussian process $\mathbf{f}_k$ and Gaussian observation noise $\mathbf{r}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{R}_k)$

$$\mathcal{L}_k = \log \mathbb{E}_{\mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)}\left[p(\mathbf{y}_k \mid \mathbf{f}_k)^\alpha\right]$$

Pseudo-code for Kalman filter EP

Likelihood $p(\mathbf{y}_k \mid \mathbf{f}_k) = h(\mathbf{f}_k, \mathbf{r}_k)$ is a nonlinear function of Gaussian process $\mathbf{f}_k$ and Gaussian observation noise $\mathbf{r}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{R}_k)$

$$\mathcal{L}_k = \log \mathbb{E}_{\mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)}\left[p(\mathbf{y}_k \mid \mathbf{f}_k)^\alpha\right]$$

Linearisation w.r.t. $\mathbf{f}_k$ and $\mathbf{r}_k$ via first-order Taylor expansion leads to a Gaussian approximation:

$$\begin{aligned} h(\mathbf{f}_k, \mathbf{r}_k) &\approx h(\mathbf{m}_k^-, \mathbf{0}) + \mathbf{J}_{\mathbf{f}_k}(\mathbf{f}_k - \mathbf{m}_k^-) + \mathbf{J}_{\mathbf{r}_k}\mathbf{r}_k \\ &= \mathrm{N}\left(\mathbf{y}_k \mid h(\mathbf{m}_k^-, \mathbf{0}) + \mathbf{J}_{\mathbf{f}_k}(\mathbf{f}_k - \mathbf{m}_k^-),\ \mathbf{J}_{\mathbf{r}_k}\mathbf{R}_k\mathbf{J}_{\mathbf{r}_k}^\top\right) \end{aligned}$$

Pseudo-code for Kalman filter EP

$$\mathcal{L}_k = \log \mathbb{E}_{\mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)} \left[\mathrm{N}\left(\mathbf{y}_k \mid h(\mathbf{m}_k^-, \mathbf{0}) + \mathbf{J}_{\mathbf{f}_k}(\mathbf{f}_k - \mathbf{m}_k^-),\ \mathbf{J}_{\mathbf{r}_k} \mathbf{R}_k \mathbf{J}_{\mathbf{r}_k}^\top \right)^{\alpha} \right]$$

Pseudo-code for Kalman filter EP

$$\mathcal{L}_k = c + \log \mathrm{N}\big(\mathbf{y}_k \mid h(\mathbf{m}_k^-, \mathbf{0}),\, \mathbf{J}_{\mathbf{f}_k} \mathbf{P}_k^- \mathbf{J}_{\mathbf{f}_k}^\top + \frac{1}{\alpha} \mathbf{J}_{\mathbf{r}_k} \mathbf{R}_k \mathbf{J}_{\mathbf{r}_k}^\top\big)$$

Pseudo-code for Kalman filter EP

for $k = 1 : T$

Kalman predict:

$$p(\mathbf{f}_k|\mathbf{y}_{1:k-1}) = \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-) = \mathrm{N}(\mathbf{A}_k\mathbf{m}_{k-1},\, \mathbf{A}_k\mathbf{P}_{k-1}\mathbf{A}_k^\top + \mathbf{Q}_k)$$

Moment match:

$$\mathcal{L}_k = c + \log \mathrm{N}\big(\mathbf{y}_k \mid h(\mathbf{m}_k^-, \mathbf{0}),\, \mathbf{J}_{\mathbf{f}_k}\mathbf{P}_k^-\mathbf{J}_{\mathbf{f}_k}^\top + \frac{1}{\alpha}\mathbf{J}_{\mathbf{r}_k}\mathbf{R}_k\mathbf{J}_{\mathbf{r}_k}^\top\big),$$

$$\mathbf{m}_k^{\text{site}} = \mathbf{m}_k^- - \left(\frac{\mathrm{d}^2\mathcal{L}_k}{\mathrm{d}{\mathbf{m}_k}^2}\right)^{-1}\frac{\mathrm{d}\mathcal{L}_k}{\mathrm{d}\mathbf{m}_k}, \quad \mathbf{P}_k^{\text{site}} = \alpha\left(-\mathbf{P}_k^- - \left(\frac{\mathrm{d}^2\mathcal{L}_k}{\mathrm{d}{\mathbf{m}_k}^2}\right)^{-1}\right).$$

Kalman update:

$$\mathbf{S}_k = \mathbf{P}_k^- + \mathbf{P}_k^{\text{site}}, \qquad \mathbf{K}_k = \mathbf{P}_k^-\mathbf{S}_k^{-1},$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k(\mathbf{m}_k^{\text{site}} - \mathbf{m}_k^-), \qquad \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top.$$

$$p(\mathbf{f}_k|\mathbf{y}_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k)$$

end for

Pseudo-code for Kalman filter EP

for $k = 1 : T$

Kalman predict:

$$p(\mathbf{f}_k|\mathbf{y}_{1:k-1}) = \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-) = \mathrm{N}(\mathbf{A}_k\mathbf{m}_{k-1},\, \mathbf{A}_k\mathbf{P}_{k-1}\mathbf{A}_k^\top + \mathbf{Q}_k)$$

Moment match:

$$\mathbf{P}_k^{\text{site}} = \left(\mathbf{J}_{\mathbf{f}_k}^\top(\mathbf{J}_{\mathbf{r}_k}\mathbf{R}_k\mathbf{J}_{\mathbf{r}_k}^\top)^{-1}\mathbf{J}_{\mathbf{f}_k}\right)^{-1},$$

$$\mathbf{m}_k^{\text{site}} = \mathbf{m}_k^- + (\mathbf{P}_k^{\text{site}} + \mathbf{P}_k^-)\mathbf{J}_{\mathbf{f}_k}^\top(\mathbf{J}_{\mathbf{r}_k}\mathbf{R}_k\mathbf{J}_{\mathbf{r}_k}^\top + \mathbf{J}_{\mathbf{f}_k}\mathbf{P}_k^-\mathbf{J}_{\mathbf{f}_k}^\top)^{-1}(\mathbf{y}_k - h(\mathbf{m}_k^-, \mathbf{0}))$$

Kalman update:

$$\mathbf{S}_k = \mathbf{P}_k^- + \mathbf{P}_k^{\text{site}}, \qquad \mathbf{K}_k = \mathbf{P}_k^-\mathbf{S}_k^{-1},$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k(\mathbf{m}_k^{\text{site}} - \mathbf{m}_k^-), \qquad \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top.$$

$$p(\mathbf{f}_k|\mathbf{y}_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k)$$

end for

Pseudo-code for Kalman filter EP

for $k = 1 : T$

Kalman predict:

$$p(\mathbf{f}_k|\mathbf{y}_{1:k-1}) = \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-) = \mathrm{N}(\mathbf{A}_k\mathbf{m}_{k-1},\ \mathbf{A}_k\mathbf{P}_{k-1}\mathbf{A}_k^\top + \mathbf{Q}_k)$$

Moment match:

Kalman update:

$$\mathbf{S}_k = \mathbf{J}_{\mathbf{f}_k}\mathbf{P}_k^-\mathbf{J}_{\mathbf{f}_k}^\top + \mathbf{J}_{\mathbf{r}_k}\mathbf{R}_k\mathbf{J}_{\mathbf{r}_k}^\top, \qquad \mathbf{K}_k = \mathbf{P}_k^-\mathbf{J}_{\mathbf{f}_k}^\top\mathbf{S}_k^{-1},$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k(\mathbf{y}_k - h(\mathbf{m}_k^-, \mathbf{0})), \qquad \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top.$$

$$p(\mathbf{f}_k|\mathbf{y}_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k)$$

end for

Pseudo-code for Kalman filter EP

for $k = 1 : T$

Kalman predict:

$$p(\mathbf{f}_k | \mathbf{y}_{1:k-1}) = \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-) = \mathrm{N}(\mathbf{A}_k \mathbf{m}_{k-1},\ \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^\top + \mathbf{Q}_k)$$

Moment match:

Kalman update:

$$\mathbf{S}_k = \mathbf{J}_{\mathbf{f}_k} \mathbf{P}_k^- \mathbf{J}_{\mathbf{f}_k}^\top + \mathbf{J}_{\mathbf{r}_k} \mathbf{R}_k \mathbf{J}_{\mathbf{r}_k}^\top, \qquad \mathbf{K}_k = \mathbf{P}_k^- \mathbf{J}_{\mathbf{f}_k}^\top \mathbf{S}_k^{-1},$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k (\mathbf{y}_k - h(\mathbf{m}_k^-, \mathbf{0})), \qquad \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^\top.$$

$$p(\mathbf{f}_k | \mathbf{y}_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k)$$

This is exactly the EKF

end for

Pseudo-code for Kalman filter EP

for $k = 1 : T$

Kalman predict:

$$p(\mathbf{f}_k | \mathbf{y}_{1:k-1}) = \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-) = \mathrm{N}(\mathbf{A}_k \mathbf{m}_{k-1},\ \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^\top + \mathbf{Q}_k)$$

Moment match:

But storing the sites allowed us to iterate

Kalman update:

$$\mathbf{S}_k = \mathbf{J}_{\mathbf{f}_k} \mathbf{P}_k^- \mathbf{J}_{\mathbf{f}_k}^\top + \mathbf{J}_{\mathbf{r}_k} \mathbf{R}_k \mathbf{J}_{\mathbf{r}_k}^\top, \qquad \mathbf{K}_k = \mathbf{P}_k^- \mathbf{J}_{\mathbf{f}_k}^\top \mathbf{S}_k^{-1},$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k(\mathbf{y}_k - h(\mathbf{m}_k^-, \mathbf{0})), \qquad \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^\top.$$

$$p(\mathbf{f}_k | \mathbf{y}_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k)$$

This is exactly the EKF

end for

Pseudo-code for Kalman filter EP

for $k = 1 : T$

Kalman predict:

$$p(\mathbf{f}_k|\mathbf{y}_{1:k-1}) = \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-) = \mathrm{N}(\mathbf{A}_k\mathbf{m}_{k-1},\, \mathbf{A}_k\mathbf{P}_{k-1}\mathbf{A}_k^\top + \mathbf{Q}_k)$$

Moment match:

$$\mathbf{P}_k^{\text{site}} = \left(\mathbf{J}_{\mathbf{f}_k}^\top(\mathbf{J}_{\mathbf{r}_k}\mathbf{R}_k\mathbf{J}_{\mathbf{r}_k}^\top)^{-1}\mathbf{J}_{\mathbf{f}_k}\right)^{-1},$$

$$\mathbf{m}_k^{\text{site}} = \mathbf{m}_k^- + (\mathbf{P}_k^{\text{site}} + \mathbf{P}_k^-)\mathbf{J}_{\mathbf{f}_k}^\top(\mathbf{J}_{\mathbf{r}_k}\mathbf{R}_k\mathbf{J}_{\mathbf{r}_k}^\top + \mathbf{J}_{\mathbf{f}_k}\mathbf{P}_k^-\mathbf{J}_{\mathbf{f}_k}^\top)^{-1}(\mathbf{y}_k - h(\mathbf{m}_k^-, \mathbf{0}))$$

Kalman update:

$$\mathbf{S}_k = \mathbf{P}_k^- + \mathbf{P}_k^{\text{site}}, \qquad \mathbf{K}_k = \mathbf{P}_k^-\mathbf{S}_k^{-1},$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k(\mathbf{m}_k^{\text{site}} - \mathbf{m}_k^-), \qquad \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top.$$

$$p(\mathbf{f}_k|\mathbf{y}_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k)$$

end for

Pseudo-code for Kalman filter EP

for $k = 1 : T$

Kalman predict:

$$p(\mathbf{f}_k|\mathbf{y}_{1:k-1}) = \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-) = \mathrm{N}(\mathbf{A}_k\mathbf{m}_{k-1},\, \mathbf{A}_k\mathbf{P}_{k-1}\mathbf{A}_k^\top + \mathbf{Q}_k)$$

Moment match:

$$\mathbf{P}_k^{\text{site}} = \left(\mathbf{J}_{\mathbf{f}_k}^\top(\mathbf{J}_{\mathbf{r}_k}\mathbf{R}_k\mathbf{J}_{\mathbf{r}_k}^\top)^{-1}\mathbf{J}_{\mathbf{f}_k}\right)^{-1},$$

$$\mathbf{m}_k^{\text{site}} = \mathbf{m}_k^- + (\mathbf{P}_k^{\text{site}} + \mathbf{P}_k^-)\mathbf{J}_{\mathbf{f}_k}^\top(\mathbf{J}_{\mathbf{r}_k}\mathbf{R}_k\mathbf{J}_{\mathbf{r}_k}^\top + \mathbf{J}_{\mathbf{f}_k}\mathbf{P}_k^-\mathbf{J}_{\mathbf{f}_k}^\top)^{-1}(\mathbf{y}_k - h(\mathbf{m}_k^-, \mathbf{0}))$$

Kalman update:

$$\mathbf{S}_k = \mathbf{P}_k^- + \mathbf{P}_k^{\text{site}}, \qquad \mathbf{K}_k = \mathbf{P}_k^-\mathbf{S}_k^{-1},$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k(\mathbf{m}_k^{\text{site}} - \mathbf{m}_k^-), \qquad \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top.$$

$$p(\mathbf{f}_k|\mathbf{y}_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k)$$

This is now a globally iterated EKF

end for

Pseudo-code for Kalman filter EP

for $k = 1 : T$

Kalman predict:

$$p(\mathbf{f}_k | \mathbf{y}_{1:k-1}) = \mathrm{N}(\mathbf{m}_k^-, \mathbf{P}_k^-) = \mathrm{N}(\mathbf{A}_k \mathbf{m}_{k-1},\ \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^\top + \mathbf{Q}_k)$$

Moment match:

$$\mathbf{P}_k^{\text{site}} = \left(\mathbf{J}_{\mathbf{f}_k}^\top (\mathbf{J}_{\mathbf{r}_k} \mathbf{R}_k \mathbf{J}_{\mathbf{r}_k}^\top)^{-1} \mathbf{J}_{\mathbf{f}_k}\right)^{-1},$$

Site updates don't depend on α

$$\mathbf{m}_k^{\text{site}} = \mathbf{m}_k^- + (\mathbf{P}_k^{\text{site}} + \mathbf{P}_k^-) \mathbf{J}_{\mathbf{f}_k}^\top (\mathbf{J}_{\mathbf{r}_k} \mathbf{R}_k \mathbf{J}_{\mathbf{r}_k}^\top + \mathbf{J}_{\mathbf{f}_k} \mathbf{P}_k^- \mathbf{J}_{\mathbf{f}_k}^\top)^{-1} (\mathbf{y}_k - h(\mathbf{m}_k^-, \mathbf{0}))$$

Kalman update:

$$\mathbf{S}_k = \mathbf{P}_k^- + \mathbf{P}_k^{\text{site}}, \qquad \mathbf{K}_k = \mathbf{P}_k^- \mathbf{S}_k^{-1},$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k (\mathbf{m}_k^{\text{site}} - \mathbf{m}_k^-), \qquad \mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^\top.$$

$$p(\mathbf{f}_k | \mathbf{y}_{1:k}) = \mathrm{N}(\mathbf{m}_k, \mathbf{P}_k)$$

This is now a globally iterated EKF

end for

Unifying EP and the EKF

- For sequential data, the EKF is equivalent to single-sweep EP where the moment matching integral is solved via linearisation.

Unifying EP and the EKF

- For sequential data, the EKF is equivalent to single-sweep EP where the moment matching integral is solved via linearisation.

- Our algorithm iteratively refines the EKF by linearising about the cavity, rather than the filter predictions (prior).

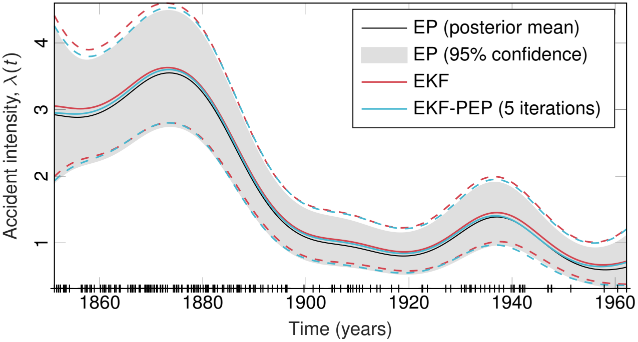
Unifying EP and the EKF

- For sequential data, the EKF is equivalent to single-sweep EP where the moment matching integral is solved via linearisation.
- Our algorithm iteratively refines the EKF by linearising about the cavity, rather than the filter predictions (prior).
- When $\alpha = 0$ we recover the Posterior Linearisation Filter. Lack of cavity calculation makes it very stable.

Unifying EP and the EKF

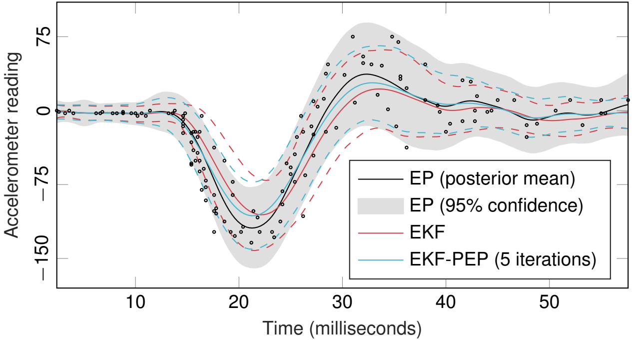
- For sequential data, the EKF is equivalent to single-sweep EP where the moment matching integral is solved via linearisation.
- Our algorithm iteratively refines the EKF by linearising about the cavity, rather than the filter predictions (prior).
- When $\alpha = 0$ we recover the Posterior Linearisation Filter. Lack of cavity calculation makes it very stable.
- Linearisation allows for straightforward calculation of cross-covariances vs. quadrature methods.

Example: log-Gaussian Cox process



The coal mining accident task (log-Gaussian Cox process) is well approximated by local linearisations, and iteration improves the match to the EP posterior.

Example: heteroscedastic noise



Linearisation in the motorcycle crash task (heteroscedastic noise) is a crude approximation, but iterating still improves the posterior.

Conclusions

- Spectral mixture GPs can be written as SDEs and have a close connection to probabilistic time-frequency analysis

- We can perform full power EP in the state space GP setting ($\mathcal{O}(n)$)

- Linearisation in state space power EP recovers the EKF and PLF (more connections to be found ...)

Thanks for listening!

contact: **william.wilkinson@aalto.fi**

Bibliography

Hartikainen, J. and Särkkä, S. (2010).
Kalman filtering and smoothing solutions to temporal Gaussian process regression models.
In *International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 379–384. IEEE.

Solin, A. and Särkkä, S. (2014).
Explicit link between periodic covariance functions and state space models.
In *Artificial Intelligence and Statistics*, pages 904–912.

Turner, R. E. and Sahani, M. (2014).
Time-frequency analysis as probabilistic inference.
IEEE Transactions on Signal Processing, 62(23):6171.

Wilkinson, W. J., Andersen, M. R., Reiss, J. D., Stowell, D., and Solin, A. (2019a).
End-to-end probabilistic inference for nonstationary audio analysis.
In *International Conference on Machine Learning*, pages 6776–6785.

Wilkinson, W. J., Chang, P. E., Andersen, M. R., and Solin, A. (2019b).
Global approximate inference via local linearisation for temporal gaussian processes.
In *Second Symposium on Advances in Approximate Bayesian Inference*.

Wilkinson, W. J., Riis Andersen, M., Reiss, J. D., Stowell, D., and Solin, A. (2019c).
Unifying probabilistic models for time-frequency analysis.
In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3352–3356.

contact: **william.wilkinson@aalto.fi**