Probabilistic Machine Learning
Lecture 21
Hidden Markov Models

Philipp Hennig

13 July 2023

Eberhard Karls Universität Tübingen

Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning

Reminder: Goal for this week – *Time Series* as a problem class.

- Application Layer: Data arriving as a stream
- Model Structure Layer: Markov Chains / Hidden Markov Models
- Concrete Model Layer: Gauss–Markov Models
- Algorithm Layer: Kalman Filter & RTS Smoother

Today:

Theory: What is the connection between Gaussian processes and Gauss–Markov Processes?

Parameters: Can we learn the parameters of a Gauss–Markov model?

Generalization: What if the world isn't Gaussian?

Recap: Markov Chains

Processes with the *state* as a local memory

$$p(f) = \mathcal{GP}(f; 0, k) \qquad p\left(\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}\right) = \mathcal{N}\left(0, \begin{bmatrix} K^{-1}_{11} & K^{-1}_{12} & 0 & 0 \\ K^{-1}_{12} & K^{-1}_{22} & K^{-1}_{23} & 0 \\ 0 & K^{-1}_{23} & K^{-1}_{33} & K^{-1}_{34} \\ 0 & 0 & K^{-1}_{34} & K^{-1}_{44} \end{bmatrix}^{-1}\right) \qquad p(\boldsymbol{y} \mid \boldsymbol{f}) = \prod_i \mathcal{N}(y_i; f_i, \sigma^2)$$

$$\begin{tikzcd} f(x_1) \arrow[r, "K^{-1}_{12}"] \arrow[d] & f(x_2) \arrow[r, "K^{-1}_{23}"] \arrow[d] & f(x_3) \arrow[r, "K^{-1}_{34}"] \arrow[d] & f(x_4) \arrow[d] \\ y_1 & y_2 & y_3 & y_4 \end{tikzcd}$$

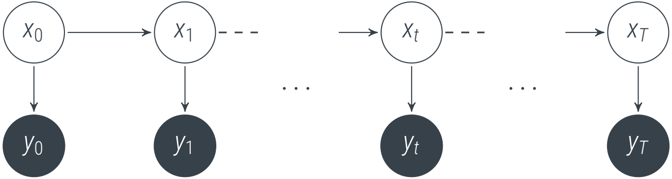
Recap: Markov Chains

Endow the algorithmic structure of filtering and smoothing

Assume:

$$p(x_t \mid X_{0:t-1}) = p(x_t \mid x_{t-1})$$

and $\quad p(y_t \mid X) = p(y_t \mid x_t)$



Filtering: $\mathcal{O}(T)$

predict: $$p(x_t \mid Y_{0:t-1}) = \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid Y_{0:t-1})\, dx_{t-1}$$ (Chapman-Kolmogorov Eq.)

update: $$p(x_t \mid Y_{0:t}) = \frac{p(y_t \mid x_t) p(x_t \mid Y_{0:t-1})}{p(y_t)}$$ (Bayes' Theorem)

Smoothing: $\mathcal{O}(T)$

smooth: $$p(x_t \mid Y) = p(x_t \mid Y_{0:t}) \int p(x_{t+1} \mid x_t) \frac{p(x_{t+1} \mid Y)}{p(x_{t+1} \mid Y_{1:t})}\, dx_{t+1}$$ (backward pass)

Time Series:

- **Markov Chains** formalize the notion of a stochastic process with a *local finite memory*
- Inference over Markov Chains separates into three operations, that can be performed in *linear* time:

Filtering: $\mathcal{O}(T)$

predict: $$p(x_t \mid Y_{0:t-1}) = \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid Y_{0:t-1})\, dx_{t-1}$$ (Chapman-Kolmogorov Eq.)

update: $$p(x_t \mid Y_{0:t}) = \frac{p(y_t \mid x_t) p(x_t \mid Y_{0:t-1})}{p(y_t)}$$ (Bayes' Theorem)

Smoothing: $\mathcal{O}(T)$

smooth: $$p(x_t \mid Y) = p(x_t \mid Y_{0:t}) \int p(x_{t+1} \mid x_t) \frac{p(x_{t+1} \mid Y)}{p(x_{t+1} \mid Y_{0:t})}\, dx_{t+1}$$ (backward pass)

procedure $\textsc{Inference}(Y, p(x_0), p(x_t \mid x_{t-1})\ \forall t, p(y_t \mid x_t)\ \forall t)$

  for i=1,…,n **do** // Filtering

    $p(x_t \mid y_{1:t-1}) = \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid Y_{0:t-1})\, dx_{t-1}$ // Chapman-Kolmogorov eq.

    $p(x_t \mid y_{1:t}) = p(y_t \mid x_t) p(x_t \mid Y_{0:t-1}) / p(y_t)$ // Update

  end for

  for i=n-1,…,0 **do** // Smoothing

    $p(x_t \mid Y) = p(x_t \mid Y_{0:t}) \int p(x_{t+1} \mid x_t) p(x_{t+1} \mid Y) p(x_{t+1} \mid Y_{1:t})\, dx_{t+1}$

  end for

  return $p(x_t \mid Y)\ \forall t = 0, \dots, n$ // return all marginals

end procedure

Gauss–Markov Models

Local structure for univariate Gaussian models

$$p(x(t_{i+1}) \mid X_{1:i}) = \mathcal{N}(x_{i+1}; Ax_i, Q) \quad \text{and} \quad p(x_0) = \mathcal{N}(x_0; m_0, P_0) \quad \text{and} \quad p(y_i \mid X) = \mathcal{N}(y_i; Hx_i, R)$$

procedure FILTER(m_0, P_0, A, Q, H, R, y)
  for $t = 1, 2, \dots, T$ **do** // $\mathcal{O}(N)$
    $m_t^- = Am_{t-1}$ // predictive mean
    $P_t^- = AP_{t-1}A^\intercal + Q$ // predictive covariance. $\mathcal{O}(|X|^3)$
    $z = y - Hm_t^-$ // residual
    $S = HP_t^- H^\intercal + R$ // innovation covariance
    $K = P_t^- H^\intercal S^{-1}$ // gain. $\mathcal{O}(|y|^3)$ Note you probably don't want to compute S^{-1} explicitly...
    $m_t = m_t^- + Kz$ // updated mean
    $P_t = (I - KH)P_t^-$ // updated covariance
  end for
  return $(m_t, P_t), (m_t^-, P_t^-)$
end procedure

The entire filtering pass through N time steps has complexity $\mathcal{O}(N \cdot (|X|^3 + |y|^3))$.

Gauss–Markov Models

Local structure for univariate Gaussian models

$$p(x(t_{i+1}) \mid X_{1:i}) = \mathcal{N}(x_{i+1}; Ax_i, Q) \quad \text{and} \quad p(x_0) = \mathcal{N}(x_0; m_0, P_0) \quad \text{and} \quad p(y_i \mid X) = \mathcal{N}(y_i; Hx_i, R)$$

```
procedure SMOOTHER(m_0, P_0, A, Q, H, R, y)
    (m_t, P_t), (m_t^-, P_t^-) ⇐ FILTER(m_0, P_0, A, Q, H, R, y)
    for t = T − 1, T − 2, ..., 1 do
        G_t = P_t A^⊺ (P_{t+1}^-)^{-1}                      // RTS gain. Complexity O(|X|^3)
        m_t^s = m_t + G_t(m_{t+1}^s − m_{t+1}^-)             // smoothed mean
        P_t^s = P_t + G_t(P_{t+1}^s − P_{t+1}^-)G^⊺          // smoothed covariance
    end for
    return (m_t^s, P_t^s)
end procedure
```

Time Series:

- **Markov Chains** formalize the notion of a stochastic process with a *local finite memory*
- Inference over Markov Chains separates into three operations, that can be performed in *linear* time.
- If all relationships are *linear* and *Gaussian*,

$$p(x(t_i) \mid x(t_{i-1})) = \mathcal{N}(x_i; Ax_{i-1}, Q) \qquad p(y_t \mid x_t) = \mathcal{N}(y_t; Hx_t, R)$$

  then inference is analytic and given by the **Kalman Filter** and the **Rauch–Tung–Striebel Smoother**.

Question 1:

Is there a continuous time limit?

Continuous Time

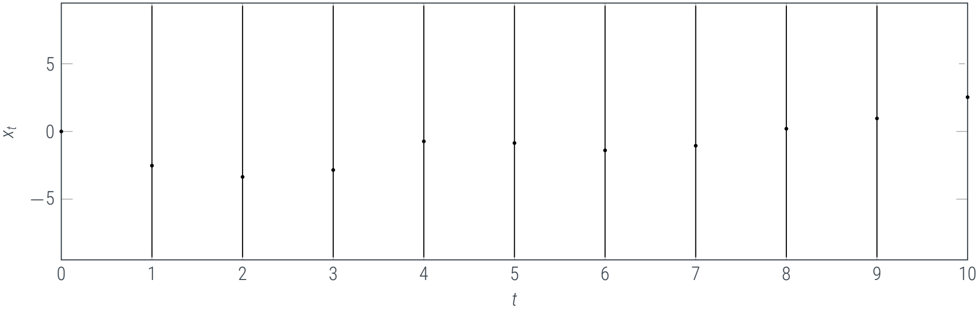
Taking one specific limit of one specific discrete model

$$p(x_{i+1} \mid x_i) \sim \mathcal{N}(x_{i+1}; x_i, Q)$$

Continuous Time

Taking one specific limit of one specific discrete model

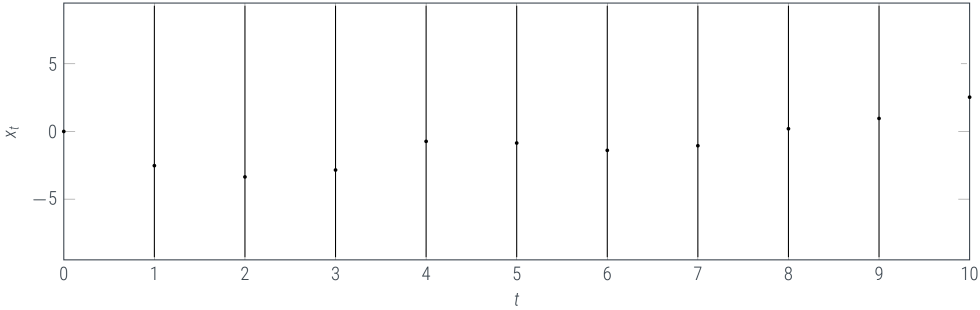
$$p(x_{i+1} \mid x_i) \sim \mathcal{N}(x_{i+1}; x_i, Q)$$



Continuous Time

Taking one specific limit of one specific discrete model

$$p(x(t+\delta t) \mid x(t)) \sim \mathcal{N}\left(x(t+\delta t); x(t), Q\right)$$

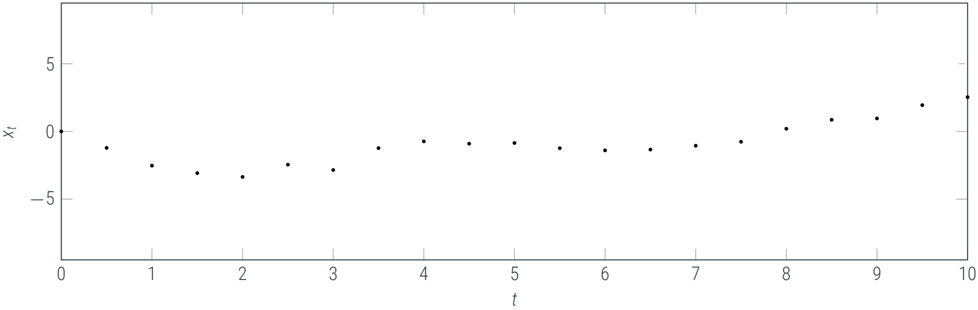


$$\delta t = 1 \qquad Q = 1$$

Continuous Time

Taking one specific limit of one specific discrete model

$$p(x(t+\delta t) \mid x(t)) \sim \mathcal{N}\left(x(t+\delta t); x(t), Q\right)$$

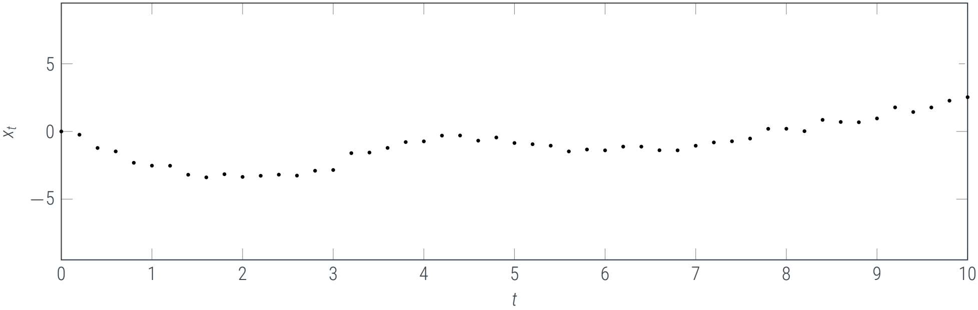


$$\delta t = 1/2 \qquad Q = 1/2$$

Continuous Time

Taking one specific limit of one specific discrete model

$$p(x(t + \delta t) \mid x(t)) \sim \mathcal{N}(x(t + \delta t); x(t), Q)$$

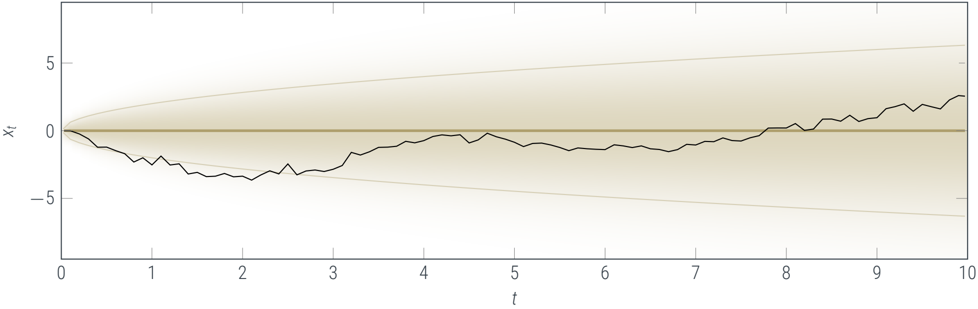


$$\delta t = 1/4 \qquad Q = \delta t$$

Continuous Time

Taking one specific limit of one specific discrete model

$$p(x(t+\delta t) \mid x(t)) \sim \mathcal{N}\left(x(t+\delta t); x(t), Q\right)$$



$$\delta t \rightarrow 0 \qquad Q_{\delta t} = ???$$

Continuous Time

The Wiener process: One specific limit of one specific discrete model

A different way to write things

$$x(t + \delta t) = x(t) + \Delta\omega(t), \qquad \text{with not-really-defined } \Delta\omega,$$

A different way to write things

$$
\begin{aligned}
x(t+\delta t) &= x(t) + \Delta\omega(t), \qquad \text{with not-really-defined } \Delta\omega, \\
\Leftrightarrow \qquad \frac{x(t+\delta t) - x(t)}{\delta t} &= \frac{\Delta\omega(t)}{\delta t}.
\end{aligned}
$$

Continuous Time

The Wiener process: One specific limit of one specific discrete model

A different way to write things

$$
\begin{aligned}
& x(t+\delta t) = x(t) + \Delta\omega(t), \qquad \text{with not-really-defined } \Delta\omega, \\
\Leftrightarrow \quad & \frac{x(t+\delta t) - x(t)}{\delta t} = \frac{\Delta\omega(t)}{\delta t}.
\end{aligned}
$$

What about the limits?

A different way to write things

$$
\begin{aligned}
x(t+\delta t) &= x(t) + \Delta\omega(t), \qquad \text{with not-really-defined } \Delta\omega, \\
\Leftrightarrow \qquad \frac{x(t+\delta t) - x(t)}{\delta t} &= \frac{\Delta\omega(t)}{\delta t}.
\end{aligned}
$$

What about the limits?

A different way to write things

$$x(t + \delta t) = x(t) + \Delta\omega(t), \qquad \text{with not-really-defined } \Delta\omega,$$

$$\Leftrightarrow \qquad \frac{x(t + \delta t) - x(t)}{\delta t} = \frac{\Delta\omega(t)}{\delta t}.$$

What about the limits?

$$\lim_{\delta t \to 0} \frac{x(t + \delta t) - x(t)}{\delta t} = \frac{dx(t)}{dt},$$

Continuous Time

The Wiener process: One specific limit of one specific discrete model

A different way to write things

$$x(t + \delta t) = x(t) + \Delta\omega(t), \qquad \text{with not-really-defined } \Delta\omega,$$

$$\Leftrightarrow \quad \frac{x(t+\delta t) - x(t)}{\delta t} = \frac{\Delta\omega(t)}{\delta t}.$$

What about the limits?

$$\lim_{\delta t \to 0} \frac{x(t+\delta t) - x(t)}{\delta t} = \frac{dx(t)}{dt},$$

$$\lim_{\delta t \to 0} \frac{\Delta\omega(t)}{\delta t} = ???$$

Continuous Time

The Wiener process: One specific limit of one specific discrete model

A different way to write things

$$x(t+\delta t) = x(t) + \Delta\omega(t), \qquad \text{with not-really-defined } \Delta\omega,$$

$$\Leftrightarrow \qquad \frac{x(t+\delta t) - x(t)}{\delta t} = \frac{\Delta\omega(t)}{\delta t}.$$

What about the limits?

$$\lim_{\delta t \to 0} \frac{x(t+\delta t) - x(t)}{\delta t} = \frac{dx(t)}{dt},$$

$$\lim_{\delta t \to 0} \frac{\Delta\omega(t)}{\delta t} = w(t), \quad \text{where} \quad w(t) \sim \mathcal{N}(0,1). \qquad \text{(“\textit{weak}” derivative)}$$

Continuous Time

The Wiener process: One specific limit of one specific discrete model

A different way to write things

$$x(t+\delta t) = x(t) + \Delta\omega(t), \qquad \text{with not-really-defined } \Delta\omega,$$
$$\Leftrightarrow \qquad \frac{x(t+\delta t) - x(t)}{\delta t} = \frac{\Delta\omega(t)}{\delta t}.$$

What about the limits?

$$\lim_{\delta t \to 0} \frac{x(t+\delta t) - x(t)}{\delta t} = \frac{dx(t)}{dt},$$
$$\lim_{\delta t \to 0} \frac{\Delta\omega(t)}{\delta t} = w(t), \quad \text{where} \quad w(t) \sim \mathcal{N}(0,1). \qquad \text{(“}\textit{weak}\text{” derivative)}$$

This is one of the key properties of the *Wiener process* (aka. *Brownian motion*).

Note that: *This is not a proper definition of the Wiener process!* This would go beyond the scope of this course. See [Särkkä & Solin, *Applied Stochastic Differential Equations*, 2019] for a thorough introduction.

Linear Time-Invariant Stochastic Differential Equations

A *very* pragmatic introduction to SDEs

For our purposes the (linear, time-invariant) **Stochastic Differential Equation (SDE)**

$$dx(t) = Fx(t)\,dt + L\,d\omega(t),$$

together with $x(0) = x_0$, describes the local behaviour of the (unique) Gaussian process with

$$\mathbb{E}(x(t)) =: m(t) = e^{Ft}x_0 \qquad \operatorname{cov}(x(t), x(t')) = \int_{\min(t,t')}^{\max(t,t')} e^{F(\max(t,t')-\tau)} L L^\intercal e^{F^\intercal(\max(t,t')-\tau)}\,d\tau$$

This GP is known as the **solution** of the SDE. It gives rise to the discrete-time stochastic recurrence relation $p(x(t_{i+1}) \mid x(t_i)) = \mathcal{N}(x(t_{i+1}); A_i x(t_i), Q_i)$ with $(\Delta t_i := t_{i+1} - t_i)$

$$A_i = e^{F\Delta t_i} \quad \text{and} \quad Q_i = \int_0^{\Delta t_i} e^{F(\Delta t_i-\tau)} L L^\intercal e^{F^\intercal(\Delta t_i-\tau)}\,d\tau.$$

Matrix exponential: $e^X := \sum_{i=0}^{\infty} \frac{X^i}{i!}$. *Thus :* $\quad e^0 = I, \quad (e^X)^{-1} = e^{-X}, \quad X = VDV^{-1} \Rightarrow Ve^DV^{-1}, \quad e^{\operatorname{diag}_i d_i} = \operatorname{diag}_i e^{d_i}, \quad \det e^X = e^{\operatorname{tr} X}.$

What this means:

- LTI-SDEs have a correspondence to GPs (so-called *Gauss–Markov processes*)
- LTI-SDEs can be discretized *exactly* to get discrete, linear Gaussian transition models

$\Rightarrow$ Gauss–Markov process inference can be done *in linear time* via filtering and smoothing!

The Connection to GPs

Some well-studied examples

$$dx(t) = Fx(t)\,dt + L\,d\omega_t$$

$$\mathbb{E}(x(t)) =: m(t) = e^{Ft}x_0 \qquad \operatorname{cov}(x(t), x(t')) =: k(t,t') = \int_{\min(t,t')}^{\max(t,t')} e^{F(\max(t,t')-\tau)}LL^\intercal e^{F^\intercal(\max(t,t')-\tau)}\,d\tau$$

The Connection to GPs

Some well-studied examples

$$dx(t) = Fx(t)\,dt + L\,d\omega_t$$

$$\mathbb{E}(x(t)) =: m(t) = e^{Ft}x_0 \quad \operatorname{cov}(x(t), x(t')) =: k(t,t') = \int_{\min(t,t')}^{\max(t,t')} e^{F(\max(t,t')-\tau)} L L^\intercal e^{F^\intercal(\max(t,t')-\tau)}\,d\tau$$

The (scaled) Wiener process

$$F = 0, L = \theta \qquad \Rightarrow \qquad m(t) = x_0 \qquad k(t,t') = \theta^2 \min(t,t')$$

$$A_i = I \qquad Q_i = \theta^2 (t_{i+1} - t_i)$$

The Connection to GPs

Some well-studied examples

$$dx(t) = Fx(t)\,dt + L\,d\omega_t$$

$$\mathbb{E}(x(t)) =: m(t) = e^{Ft}x_0 \qquad \operatorname{cov}(x(t), x(t')) =: k(t,t') = \int_{\min(t,t')}^{\max(t,t')} e^{F(\max(t,t')-\tau)} L L^\intercal e^{F^\intercal(\max(t,t')-\tau)}\,d\tau$$

The Ornstein-Uhlenbeck process

$$F = -\frac{1}{\lambda}, L = \frac{2\theta}{\sqrt{\lambda}} \qquad \Rightarrow \qquad m(t) = x_0 e^{-\frac{t}{\lambda}} \qquad k(t,t') = \theta^2\left(e^{-\frac{|t-t'|}{\lambda}} - e^{-\frac{t+t'}{\lambda}}\right)$$

$$A_i = e^{-\Delta t_i/\lambda} \qquad Q_i = \theta^2\left(1 - e^{-2\Delta t_i/\lambda}\right)$$

$$dx(t) = Fx(t)\,dt + L\,d\omega_t$$

- So far, we have seen examples with $x(t) \in \mathbb{R}$.
- But F and L can also be matrices. Consider the example

$$x = \begin{bmatrix} x_{(1)} \\ x_{(2)} \end{bmatrix} \qquad F = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad L = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

That is:

$$\begin{bmatrix} dx_{(1)}(t) \\ dx_{(2)}(t) \end{bmatrix} = \begin{bmatrix} x_{(2)}(t)\,dt + 0\,d\omega \\ 0\,dt + d\omega \end{bmatrix} \qquad \Rightarrow \qquad x_{(1)}(t) = \int_0^t x_{(2)}(t)\,dt + [x_0]_1$$

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

$$dx(t) = Fx(t)\,dt + L\,d\omega_t$$

The Wiener velocity (aka. "once-integrated Wiener process")

$$F = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, L = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \Rightarrow \qquad m(t) = e^{Ft}x_0 \qquad k(t,t') = \frac{\min^3(t,t')}{3} + |t-t'|\frac{\min^2(t,t')}{2}$$

$$A_i = \begin{bmatrix} 1 & \Delta t_i \\ 0 & 1 \end{bmatrix}, \qquad Q_i = \begin{bmatrix} \frac{\Delta t_i^3}{3} & \frac{\Delta t_i^2}{2} \\ \frac{\Delta t_i^2}{2} & \Delta t_i \end{bmatrix}$$

Q1 summary:

- Certain Gaussian processes can be written as LTI-SDEs
  - (integrated) Wiener process
  - (integrated) Ornstein–Uhlenbeck process
  - Matern processes
  - Even the square-exponential kernel can be approximated by an LTI-SDE
- LTI-SDEs can be discretized *exactly* to get discrete, linear Gaussian transition models
- Inference in linear Gauss–Markov models (and thus in Gauss–Markov processes) can be done *in linear time* via filtering and smoothing

Question 2:

Can we learn the model?

Recap (Lecture 10) **Learning** the Kernel for GPs

Bayesian Hierarchical Inference

$$p(f \mid \theta) = \mathcal{GP}(f; m_\theta, k_\theta) \qquad \text{e.g.} \quad m_\theta(\bullet) = \phi(\bullet)^\intercal \boldsymbol{\theta}, \text{ or } k_\theta(\bullet, \circ) = \theta_1 \exp\left(-\frac{(\bullet - \circ)^2}{2\theta_2^2}\right).$$

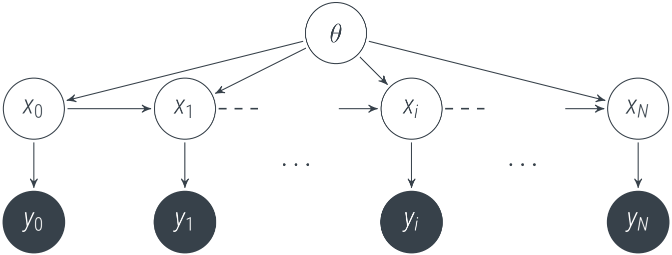
- The **evidence** in Bayes' theorem is the **marginal likelihood for the model**

$$p(f \mid \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{y} \mid f, \boldsymbol{x}, \boldsymbol{\theta}) p(f \mid, \boldsymbol{\theta})}{\int p(\boldsymbol{y} \mid f, \boldsymbol{x}, \boldsymbol{\theta}) p(f \mid, \boldsymbol{\theta})\, df} = \frac{p(\boldsymbol{y} \mid f, \boldsymbol{x}, \boldsymbol{\theta}) p(f \mid, \boldsymbol{\theta})}{p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})}$$

- For Gaussians *and Gaussian processes*, the evidence has **analytic form**:

$$\underbrace{\mathcal{N}(\boldsymbol{y}; \phi_X^{\boldsymbol{\theta}\intercal} w + \boldsymbol{b}, \Lambda)}_{p(\boldsymbol{y}|f,\boldsymbol{x},\boldsymbol{\theta})} \cdot \underbrace{\mathcal{N}(w, \mu, \Sigma)}_{p(f)} = \underbrace{\mathcal{N}(w; m_{\text{post}}^{\boldsymbol{\theta}}, V_{\text{post}}^{\boldsymbol{\theta}})}_{p(f|\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})} \cdot \underbrace{\mathcal{N}(\boldsymbol{y}; \phi_X^{\boldsymbol{\theta}\intercal} \mu + b, \phi_X^{\boldsymbol{\theta}\intercal} \Sigma \phi_X^{\boldsymbol{\theta}} + \Lambda)}_{p(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{x})}$$

$$\mathcal{N}(\boldsymbol{y}; f^{\boldsymbol{\theta}}(X), \Lambda^{\boldsymbol{\theta}}) \cdot \mathcal{GP}(f, \mu^{\boldsymbol{\theta}}, k^{\boldsymbol{\theta}}) = \mathcal{GP}(f; m_{\text{post}}^{\boldsymbol{\theta}}, V_{\text{post}}^{\boldsymbol{\theta}}) \cdot \mathcal{N}(\boldsymbol{y}; \mu^{\boldsymbol{\theta}}(X), \Lambda^{\boldsymbol{\theta}} + k^{\boldsymbol{\theta}}(X, X))$$



For Gauss–Markov Models, is there a way to compute the model evidence in $\mathcal{O}(N)$?

- Assume unknown **model hyper-parameters** θ (define $x_{-1} = \varnothing$):

$$p(\mathbf{y}, \mathbf{x}, \theta) = p(\theta) \cdot p(\mathbf{y}, \mathbf{x} \mid \theta) = p(\theta) \prod_{i=0}^{N} p(x_i \mid x_{i-1}, \theta) p(y_i \mid x_i, \theta)$$

- to learn θ, we need the **evidence** (aka. **marginal/type-II likelihood**) (define $y_{-1} = \varnothing$)

$$p(\mathbf{y} \mid \theta) = \prod_{i=0}^{N} p(y_i \mid y_{0:i-1}, \theta)$$

- the terms in the product decompose into local predictions:

$$p(y_i \mid y_{0:i-1}, \theta) = \int p(y_i, x_i \mid y_{0:i-1}, \theta)\, dx_i \qquad = \int p(y_i \mid x_i, y_{0:i-1}, \theta) p(x_i \mid y_{0:i-1}, \theta)\, dx_i$$

$$= \int p(y_i \mid x_i, \theta) p(x_i \mid y_{0:i-1}, \theta)\, dx_i$$ which, for linear Gaussian systems, is

$$= \int \mathcal{N}(y_i; Hx_i, R) \mathcal{N}(x_i; m_i^-, P_i^-)\, dx_i = \mathcal{N}(y_i; Hm_i^-, HP_i^- H^\intercal + R) = \mathcal{N}(z_i; 0, S_i)$$

Learning parameters of a Gauss–Markov Model

For the LTI system

$$p(x_i \mid x_{i-1}, \theta) = \mathcal{N}(x_i; Ax_{i-1}, Q), \quad \text{and} \quad p(y_i \mid x_i, \theta) = \mathcal{N}(y_i; Hx_i, R),$$

the (log) evidence is given by

$$\begin{aligned} p(\boldsymbol{y} \mid \theta) &= \prod_{i=1}^{N} p(y_i; y_{0:i-1}, \theta) \\ &= \prod_{i=0}^{N} \mathcal{N}(y_i; Hm_i^-, HP_i^- H^\intercal + R) \\ \log p(\boldsymbol{y} \mid \theta) &= -\frac{1}{2} \sum_{i=1}^{N} \left(z_i^\intercal S_i^{-1} z_i + \log |S_i| + \log 2\pi \right) \end{aligned}$$

In principle, this could also be used to learn A, Q, R, H directly, but there's a more elegant way to do this for linear Gaussian systems. For more, cf. Ghahramani & Hinton, 1996, and Särkkä 2013.

Question 3:

What if the world is not linear Gaussian?

Hidden Markov Models

Generalization to non-Gaussian relationships

Name	Distribution	Algorithm
Markovian System:	$p(\boldsymbol{y}, \boldsymbol{x}) = \prod_{i=0}^{N} p(x_i \mid x_{i-1}) p(y_i \mid x_i)$	General Bayesian filtering and smoothing
Linear Gaussian System:	$p(\boldsymbol{y}, \boldsymbol{x}) = \prod_{i=0}^{N} \mathcal{N}(x_i; A_i x_{i-1}, Q_i) \mathcal{N}(y_i; H x_i, R)$	Kalman filter, Rauch−Tung−Striebel smoother
Nonlinear Gaussian System:	$p(\boldsymbol{y}, \boldsymbol{x}) = \prod_{i=0}^{N} \mathcal{N}(x_i; a(x_{i-1}), Q_i) \mathcal{N}(y_i; h(x_i), R)$	e.g. Extended/Unscented/Particle filter etc.
Non-Gaussian observations:	$p(\boldsymbol{y}, \boldsymbol{x}) = \prod_{i=0}^{N} \mathcal{N}(x_i; A_i x_{i-1}, Q_i) p(y_i \mid h(x_i))$	
Hidden Markov Model (e.g.):	$p(\boldsymbol{y}, \boldsymbol{x}) = \prod_{i=0}^{N} p(x_i = \Pi x_{i-1}) \mathcal{N}(y_i; h(x_i), R)$	

- For continuous systems with nonlinear dynamics and/or non-linear observations, a number of *approximately Gaussian filters* have been developed. For more see, e.g.
  - Simo Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013 `https://users.aalto.fi/~ssarkka/pub/cup_book_online_20131111.pdf`

Because streaming data is a common data type, time series are an entire sub-field of their own, studied in a diverse range of domains. There is no time to cover them all in this course.

Summary:

Markov Chains capture **finite memory** of a time series through conditional independence

Gauss–Markov models map this state to linear algebra

Kalman filter is the name for the corresponding algorithm

SDEs (Stochastic Differential Equations) are the continuous-time limit of discrete-time stochastic recurrence relations (in particular, linear SDEs are the continuous-time generalization discrete-time linear Gaussian systems)

Parameters of the model can be learnt by optimizing the (log) evidence, which is also $\mathcal{O}(N)$.

Non-Gaussian models can be learnt by approximate inference, analogous to GP models.

Please cite this course, as

```
@techreport{Tuebingen_ProbML23 ,
    title =
    {Probabilistic Machine Learning},
    author = {Hennig, Philipp},
    series = {Lecture Notes
        in Machine Learning},
    year = {2023},
    institution = {Tübingen AI Center}}
```

Gauss–Markov models form the algorithmic scaffold for time-series models .