Sequential Monte Carlo methods

Lecture 2 – Probabilistic modelling of dynamical systems

Thomas Schön, Uppsala University

2017-08-24

Outline – Lecture 2

Aim: Explain how latent variables and Markov chains are used in probabilistic modelling of dynamical system.

Outline:

1. State space model (SSM)
2. Linear Gaussian state space model (LG-SSM)
3. Nonlinear state space model
4. Nonlinear filtering problem and its conceptual solution

Latent variable model

Model variables that are not observed are called **latent** (a.k.a. hidden, missing and unobserved) variables.

The idea of introducing latent variables into models is probably one of the most powerful concepts in probabilistic modelling.

Latent variables provide **more expressive** models that can capture **hidden structures** in data that would otherwise not be possible.

Cost: Learning the model becomes (significantly) harder.

Markov chain

The Markov chain is a probabilistic model that is used for modelling a sequence of states $(X_0, X_1, \dots, X_T)$.

Definition (Markov chain)

A stochastic process $\{X_t\}_{t\geq 0}$ is referred to as a Markov chain if, for every $k > 0$ and t,

$$p(x_{t+k} \mid x_0, x_1, \dots, x_t) = p(x_{t+k} \mid x_t).$$

A **Markov chain** is completely specified by:

1. An initial value X_0 and
2. a transition model (kernel) $\kappa(x_{t+1} \mid x_t)$ describing the transition from state X_t to state X_{t+1}.

The **state** acts as a **memory** containing all information there is to know about the phenomenon at this point in time.

Markov chain

Our two most important applications of Markov chains in this course are:

1. The Markov model is used in the **state space model (SSM)** where we can only observe the state indirectly via a measurement that is related to the state.
2. The Markov chain constitutes the basic ingredient in the Markov chain Monte Carlo (MCMC) methods.

Linear Gaussian state space model (LG-SSM)

The linear Gaussian state space model (LG-SSM) is given by

$$
\begin{aligned}
X_t &= AX_{t-1} + Bu_t + V_t, \\
Y_t &= CX_t + Du_t + E_t,
\end{aligned}
$$

where $X_t \in \mathbb{R}^{n_x}$ denotes the state, $u_t \in \mathbb{R}^{n_u}$ denotes an explanatory variable (known signal) and $Y_t \in \mathbb{R}^{n_y}$ denotes the measurement (data).

The initial state and the noise are distributed according to

$$
\begin{pmatrix} X_0 \\ V_t \\ E_t \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} P_0 & 0 & 0 \\ 0 & Q & S \\ 0 & S^{\mathsf{T}} & R \end{pmatrix} \right)
$$

Gaussian (normal) random variables

The PDF of a Gaussian variable is denoted $\mathcal{N}(x \mid \mu, \Sigma)$, i.e.,

$$
\mathcal{N}(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}(x-\mu)^{\mathsf{T}}\Sigma^{-1}(x-\mu) \right)
$$

See the appendix of the lecture notes for the basic theorems needed in manipulating Gaussian random variables.

Nonlinear state space model (SSM)

Nonlinear state space model (SSM)

The state space model (SSM) is a **Markov** chain that makes use of a **latent** variable representation to describe dynamical phenomena.

It consists of two stochastic processes:

1. unobserved (state) process $\{X_t\}_{t\geq 0}$ modelling the dynamics,
2. observed process $\{Y_t\}_{t\geq 1}$ modelling the measurements and their relationship to the unobserved state process.

$$X_t = f(X_{t-1}, \theta) + V_t,$$
$$Y_t = g(X_t, \theta) + E_t,$$

where $\theta \in \mathbb{R}^{n_\theta}$ denotes static model parameters.

The SSM offers a practical representation not only for **modelling**, but also for **reasoning** and **inference**.

Ex. – "what are X_t, θ and Y_t"?

Aim (motion capture): Compute X_t (position and orientation of the different body segments) of a person (θ describes the body shape) moving around indoors using measurements Y_t (accelerometers, gyroscopes and ultrawideband).

Show movie!

Manon Kok, Jeroen D. Hol and Thomas B. Schön. **Using inertial sensors for position and orientation estimation**, *Foundations and Trends of Signal Processing*, 2017 (to appear).

Representing the SSM using distributions

Representation using probability distributions

$$X_t \mid (X_{t-1} = x_{t-1}, \theta = \theta) \sim p(x_t \mid x_{t-1}, \theta),$$
$$Y_t \mid (X_t = x_t, \theta = \theta) \sim p(y_t \mid x_t, \theta),$$
$$X_0 \sim p(x_0 \mid \theta).$$

The unknown parameters can be modelled as either

1. deterministic but unknown (maximum likelihood) or
2. random variables (Bayesian), $\theta \sim p(\theta)$.

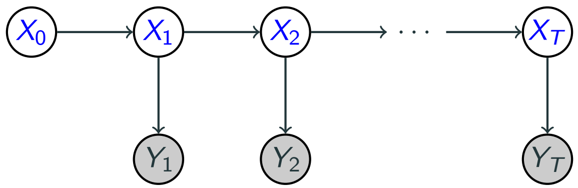
State inference: Learn about the state from the observations.

Parameter inference: Learn the (static) parameters from the observations.

The nonlinear SSM is just a special case...

A **graphical model** is a probabilistic model where a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents the conditional independency structure between random variables,

1. a set of **vertices** $\mathcal{V}$ (nodes) represents the random variables
2. a set of **edges** $\mathcal{E}$ containing elements $(i, j) \in \mathcal{E}$ connecting a pair of nodes $(i, j) \in \mathcal{V}$
3. The arrows pointing to a certain node encodes which variables the corresponding node are conditioned upon.



The nonlinear SSM is just a special case...

Representation using probabilistic program

```
x[1] ~ Gaussian(0.0, 1.0);
y[1] ~ Gaussian(x[1], 1.0);
for (t in 2..T) {
  x[t] ~ Gaussian(a*x[t - 1], 1.0);
  y[t] ~ Gaussian(x[t], 1.0);
}
```

$p(x_1)$

$p(y_1 \mid x_1)$

$p(x_t \mid x_{t-1})$

$p(y_t \mid x_t)$

Recall: A **probabilistic program** encodes a **probabilistic model** (here an LG-SSM) according to the semantics of a particular probabilistic programming language (here Birch).

Outlook – Gaussian process SSM

The Gaussian process (GP) is a **non-parametric** and **probabilistic** model for nonlinear functions.

Non-parametric means that it does not rely on any particular parametric functional form to be postulated.

$$X_t = f(X_{t-1}) + V_t, \qquad \text{s.t.} \quad f(X) \sim \mathcal{GP}(0, \kappa_{\eta,f}(x, x')),$$
$$Y_t = g(X_t) + E_t, \qquad \text{s.t.} \quad g(X) \sim \mathcal{GP}(0, \kappa_{\eta,g}(x, x')).$$

The model functions f and g are assumed to be realizations from Gaussian process priors and $V_t \sim \mathcal{N}(0, Q)$, $E_t \sim \mathcal{N}(0, R)$.

Task: Compute the posterior $p(f, g, Q, R, \eta, x_{0:T} \mid y_{1:T})$.

Roger Frigola, Fredrik Lindsten, Thomas B. Schön, and Carl Rasmussen. **Bayesian inference and learning in Gaussian process state-space models with particle MCMC.** *NIPS*, 2013.

Andreas Svensson and Thomas B. Schön. **A flexible state space model for learning nonlinear dynamical systems.** *Automatica*, 80:189-199, June, 2017.

SSM – full probabilistic model

The **full probabilistic model** is given by

$$p(x_{0:T}, \theta, y_{1:T}) = \underbrace{p(y_{1:T} \mid x_{0:T}, \theta)}_{\text{data distribution}} \underbrace{p(x_{0:T}, \theta)}_{\text{prior}}$$

Distribution describing a parametric nonlinear SSM

$$p(x_{0:T}, \theta, y_{1:T}) = \underbrace{\prod_{t=1}^{T} \underbrace{p(y_t \mid x_t, \theta)}_{\text{observation}}}_{\text{data distribution}} \underbrace{\prod_{t=1}^{T} \underbrace{p(x_t \mid x_{t-1}, \theta)}_{\text{dynamics}} \underbrace{p(x_0 \mid \theta)}_{\text{state}} \underbrace{p(\theta)}_{\text{param.}}}_{\text{prior}}$$

Model = probability distribution!

Nonlinear filtering problem

State inference

State inference refers to the problem of learning about the state $X_{k:l}$ based on the available measurements $Y_{1:t} = y_{1:t}$.

We will represent this information using **PDFs**.

Name	Probability density function
Filtering	$\mathbf{p(x_t \mid y_{1:t})}$
Joint filtering	$p(x_{0:t} \mid y_{1:t}),\ t = 1, 2, \ldots$
Prediction	$p(x_{t+1} \mid y_{1:t})$
Joint smoothing	$p(x_{1:T} \mid y_{1:T})$
Marginal smoothing	$p(x_t \mid y_{1:T}),\ t \leq T$

The nonlinear filtering problem

State filtering problem: Learn about the current state X_t based on the available measurements $Y_{1:t} = y_{1:t}$ when

$$\begin{aligned} X_t \mid (X_{t-1} = x_{t-1}) &\sim p(x_t \mid x_{t-1}), \\ Y_t \mid (X_t = x_t) &\sim p(y_t \mid x_t), \\ X_0 &\sim p(x_0). \end{aligned} \qquad \begin{aligned} X_t &= f(X_{t-1}) + V_t, \\ Y_t &= g(X_t) + E_t, \\ X_0 &\sim p(x_0). \end{aligned}$$

Strategy: Compute the filter PDF $p(x_t \mid y_{1:t})$ as accurately as possible.

Filtering problem – conceptual solution

The **measurement update**

$$p(x_t \mid y_{1:t}) = \frac{\overbrace{p(y_t \mid x_t)}^{\text{measurement}}\ \overbrace{p(x_t \mid y_{1:t-1})}^{\text{prediction pdf}}}{p(y_t \mid y_{1:t-1})},$$

and the **time update**

$$p(x_t \mid y_{1:t-1}) = \int \underbrace{p(x_t \mid x_{t-1})}_{\text{dynamics}}\ \underbrace{p(x_{t-1} \mid y_{1:t-1})}_{\text{filtering pdf}} \mathrm{d}x_{t-1}.$$

Alternatively we can of course combine the two

$$p(x_t \mid y_{1:t}) = \frac{p(y_t \mid x_t) \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) \mathrm{d}x_{t-1}}{p(y_t \mid y_{1:t-1})}.$$

No closed-form solutions available except for a few special cases.

Explicit filtering solution for LG-SSM – Kalman filter

Linear transformations of Gaussian r.v. remain Gaussian and hence completely characterized by their mean and covariance.

Measurement update

$$\begin{aligned} p(x_t \mid y_{1:t}) &= \mathcal{N}\left(x_t \,\middle|\, \widehat{x}_{t\,|\,t}, P_{t\,|\,t}\right), \\ \widehat{x}_{t\,|\,t} &= \widehat{x}_{t\,|\,t-1} + K_t \left(y_t - C\widehat{x}_{t\,|\,t-1} - Du_t\right), \\ P_{t\,|\,t} &= (I - K_t C) P_{t\,|\,t-1}, \\ K_t &= P_{t\,|\,t-1} C^\mathsf{T} \left(C P_{t\,|\,t-1} C^\mathsf{T} + R\right)^{-1}. \end{aligned}$$

Time update

$$\begin{aligned} p(x_{t+1} \mid y_{1:t}) &= \mathcal{N}\left(x_{t+1} \,\middle|\, \widehat{x}_{t+1\,|\,t}, P_{t+1\,|\,t}\right), \\ \widehat{x}_{t+1\,|\,t} &= A\widehat{x}_{t\,|\,t} + Bu_t, \\ P_{t+1\,|\,t} &= AP_{t\,|\,t}A^\mathsf{T} + Q. \end{aligned}$$

Backward computations – (too) brief

In off-line situations it often makes sense to also propagate the information backwards in time from $t = T$ to $t = 0$.

Joint smoothing PDF

$$p(x_{0:T} \mid y_{1:T}) = \prod_{t=0}^{T-1} \underbrace{p(x_t \mid x_{t+1}, y_{1:T})}_{\text{backward kernel}} p(x_T \mid y_{1:T}),$$

where

$$p(x_t \mid x_{t+1}, y_{1:T}) = \frac{p(x_{t+1} \mid x_t) p(x_t \mid y_{1:t})}{p(x_{t+1} \mid y_{1:t})}.$$

Marginal smoothing PDF

$$p(x_t \mid y_{1:T}) = p(x_t \mid y_{1:t}) \int \frac{p(x_{t+1} \mid x_t) p(x_{t+1} \mid y_{1:T})}{p(x_{t+1} \mid y_{1:t})} \mathrm{d}x_{t+1}.$$

Fredrik Lindsten and Thomas B. Schön. **Backward simulation methods for Monte Carlo statistical inference.** *Foundations and Trends in Machine Learning*, 6(1):1-143, 2013.

A few concepts to summarize lecture 2

Latent variable model: A model containing unknown variables that are not directly observed.

Markov chain: Described by an initial state and a transition kernel describing the transition from the present state to the next.

Spate space model (SSM): A latent variable model, where the latent variable (the state) is observed indirectly.

State inference: Learn about the state $X_{k:l}$ based on the available measurements $Y_{1:t} = y_{1:t}$.

Parameter inference: Learn the (static) parameters θ based on the available measurements $y_{1:T} = \{y_1, y_2, \dots, y_T\}$.

Filtering problem: Learn about the current state X_t based on the available measurements $Y_{1:t} = y_{1:t}$ by computing $p(x_t \mid y_{1:t})$.

Kalman filter: Explicit solution to the state filtering problem when the SSM is linear and Gaussian.