In this set of lecture notes, we discuss probabilistic models for time series data. We begin in Section 1 with a brief introduction to Bayesian networks and, specifically, probabilistic state-space models (SSM). We then discuss the problem of posterior inference in probabilistic SSMs, specifically the forward-backward algorithm for HMMs (Section 2) and the Rauch-Tung-Striebel smoother for linear Gaussian state-space models (Section 3). We conclude in Section 4 by briefly discussing parameter estimation via maximum likelihood estimation and the expectation-maximization (EM) algorithm.

1 Background: Bayesian networks

We begin by providing a brief introduction to Bayesian networks. Bayesian networks constitute a subclass of the larger class of probabilistic graphical models (PGMs). This framework, specifically the conditional independence properties of Bayesian networks, will be useful for our discussion of probabilistic state-space models.

Definition 1.1 (Bayesian network)**.** Consider a directed graph $\mathcal{G} = (V, E)$. Define the parent set of node i, written $\text{pa}(i) \subseteq V$, to be the collection of nodes $j \neq i$ for which there exists an edge from j to i in the graph $\mathcal{G}$. Note that $\text{pa}(i)$ can be empty.

A **Bayesian network** (BN) is a directed, acyclic graph $\mathcal{G} = (V, E)$ together with

(i) A collection of random variables $(X_i)_{i \in V}$ corresponding to each node.

(ii) A conditional probability distribution for each node $i \in V$, which depends on its parent nodes: $p(X_i = x_i | X_{\text{pa}(i)} = x_{\text{pa}(i)})$.

One can see via an induction argument that our definition of a BN yields a well-defined probability distribution over V.

The case of $|V| = 1$ is immediate. For $|V| > 1$, there exists a node i' which is not the parent of any other node $j \neq i'$; otherwise the graph must be cyclic. By assumption, the subgraph that does not include i' is a Bayesian network. Moreover, by the induction hypothesis, this BN defines a probability distribution $p(X_{-i'} = x_{-i'})$ over vertices $V \setminus \{i'\}$. Define a probability distribution over V according to $p(X_{i'} = x_{i'}, X_{i'} = x_{i'}) = p(X_{i'} = x_{i'} | X_{\text{pa}(i')} = x_{\text{pa}(i')}) p(X_{-i'} = x_{-i'})$. We notice that

$$\begin{aligned}\sum_{x_{i'}} p(X_{i'} = x_{i'} | X_{\text{pa}(i')} = x_{\text{pa}(i')}) = p(X_{-i'} = x_{-i'}) \sum_{x_{i'}} p(X_{i'} = x_{i'} | X_{\text{pa}(i')} = x_{\text{pa}(i')}) \\ = p(X_{-i'} = x_{-i'})\end{aligned}$$

Since $\sum_{x_{-i'}} p(X_{-i'} = x_{-i'}) = 1$, this implies $p(X_{i'} = x_{i'}, X_{i'} = x_{i'})$ is a well-defined probability distribution. □

The induction step gives us a useful representation for the joint probability distribution defined by the Bayesian network:

$$p(X = x) = \prod_{i \in V} p(X_i = x_i | X_{\text{pa}(i)} = x_{\text{pa}(i)}). \tag{1}$$

In other words, the probability of observing an outcome x is equal to the product of the conditional probabilities of each child node i being equal to x_i conditional on the value of its parents $x_{\text{pa}(i)}$.

1.1 Dependency relationships in Bayesian networks

Bayesian networks are nice probabilistic models to work with since they tell us about the dependencies between random variables $(X_i)_{i\in V}$ defined by the nodes in the graph. Below we summarize one such dependency:

Proposition 1.2. *Let $\mathcal{G}$ be a Bayesian network. Suppose $k \in pa(j)$ and $j \in pa(i)$. Moreover, suppose that removing node j disconnects the graph into two components A and B with $i \in A$ and $k \in B$. Then*

$$p(X_i = x_i, X_k = x_k | X_j = x_j) = p(X_i = x_i | X_j = x_j) p(X_k = x_k | X_j = x_j)$$

i.e., X_i and X_k are conditionally independent given X_j. However, generally

$$p(X_i = x_i, X_k = x_k) \neq p(X_i = x_i) p(X_k = x_k)$$

i.e., X_i and X_k are not unconditionally independent.

Proof. Write the joint distribution of $(X_v)_{v\in V}$ as

$$p(X = x) = \prod_{v\in V} p(x_v | x_{\mathrm{pa}(v)}). \tag{2}$$

We rewrite (2) as

$$p(X = x) = \underbrace{\left(\prod_{a\in A} p(X_a = x_a | X_{\mathrm{pa}(a)} = x_{\mathrm{pa}(a)}) \right)}_{\Phi_A(x_j, x_A)} \underbrace{p(X_j = x_j | X_{\mathrm{pa}(j)} = x_{\mathrm{pa}(j)})}_{\phi_B(x_j, x_B)} \underbrace{\left(\prod_{b\in B} p(X_b = x_b | X_{\mathrm{pa}(b)} = x_{\mathrm{pa}(b)}) \right)}_{\Phi_B(x_j, x_B)}.$$

By our assumption, Φ_A is a well-defined function of x_j and x_A, whereas Φ_B and ϕ_B are well-defined functions of x_j and x_B.

From here, we sum over variables $V \setminus \{i, j, k\}$ to compute the joint distribution $p(X_i, X_j, X_k)$

$$\begin{aligned} &p(X_i = x_i, X_j = x_j, X_k = x_k) = F_{ij}(x_i, x_j) F_{jk}(x_j, x_k), \\ &F_{ij}(x_i, x_j) = \sum_{x_{A\setminus\{i\}}} \Phi_A(x_j, x_A), \quad F_{jk}(x_j, x_k) = \sum_{x_{B\setminus\{k\}}} \Phi_A(x_j, x_B). \end{aligned} \tag{3}$$

Moreover, we compute

$$\begin{aligned} p(X_i = x_i, X_j = x_j) &= F_{ij}(x_i, x_j) \left(\sum_{x_k} F_{jk}(x_j, x_k) \right), \\ p(X_j = x_j, X_k = x_k) &= F_{jk}(x_j, x_k) \left(\sum_{x_i} F_{ij}(x_i, x_j) \right), \\ p(X_j = x_j) &= \left(\sum_{x_i} F_{ij}(x_i, x_j) \right) \left(\sum_{x_k} F_{jk}(x_j, x_k) \right). \end{aligned} \tag{4}$$

Combining equations (3) and (4) yields an expression for $p(X_i = x_i, X_k = x_k | X_j = x_j)$.

$$p(X_i = x_i, X_k = x_k | X_j = x_j) = \frac{F_{ij}(x_i, x_j) F_{jk}(x_j, x_k)}{\left(\sum_{x_i} F_{ij}(x_i, x_j)\right) \left(\sum_{x_k} F_{jk}(x_j, x_k)\right)}$$

$$= \frac{F_{ij}(x_i, x_j)}{\left(\sum_{x_i} F_{ij}(x_i, x_j)\right)} \frac{F_{jk}(x_j, x_k)}{\left(\sum_{x_k} F_{jk}(x_j, x_k)\right)}$$
$$= p(X_i = x_i | X_j = x_j) p(X_k = x_k | X_j = x_j).$$

□

We note that our condition on nodes i and k is sufficient, but not necessary to ensure conditional independence. For a simple counterexample, consider a four node BN $V = \{i, j, k, w\}$ in which the edges are $k \in \text{pa}(j)$, $j \in \text{pa}(i)$, $i \in \text{pa}(w)$, $j \in \text{pa}(w)$. Removing node j does not yield two disconnected components, since i and j are both parents of w. However, a short calculation reveals

$$\begin{aligned}
&p(X_i = x_i, X_k = x_k | X_j = x_j) \\
&= \frac{p(X_k = x_k) p(X_i = x_i | X_j = x_j) p(X_j = x_j | X_k = x_k) \sum_{x_w} p(X_w = x_w | X_i = x_i, X_k = x_k)}{\sum_{x_i', x_k', x_w} p(X_k = x_k') p(X_i = x_i' | X_j = x_j) p(X_j = x_j | X_k = x_k') p(X_w = x_w | X_i = x_i', X_k = x_k')} \\
&\overset{(\star)}{=} \frac{p(X_k = x_k) p(X_i = x_i | X_j = x_j) p(X_j = x_j | X_k = x_k)}{\left(\sum_{x_k'} p(X_k = x_k') p(X_j = x_j | X_k = x_k')\right)\left(\sum_{x_i'} p(X_i = x_i' | X_j = x_j)\right)} \\
&\overset{(\star\star)}{=} \frac{p(X_k = x_k) p(X_j = x_j | X_k = x_k)}{p(X_j = x_j)} p(X_i = x_i | X_j = x_j) \\
&= p(X_k = x_k | X_j = x_j) p(X_i = x_i | X_j = x_j).
\end{aligned}$$

For $(\star)$ we used $\sum_{x_w} p(X_w = x_w | X_i = x_i', X_k = x_k') = 1$ for all x_i', x_k'; and for $(\star\star)$ we used $\sum_{x_i'} p(X_{i'} = x_{i'} | X_j = x_j') = 1$ for all x_j'.

In other words, X_i and X_k are conditionally independent given X_j. However, if we were to additionally condition on X_w, the result is no longer true i.e., X_i and X_k are not conditionally independent, given X_j and X_w. Intuitively, this is because both X_i and X_k are parents of X_w, and knowing the value of X_w tells us about the values of its parents. For a more thorough treatment of dependency relationships in a Bayesian network, see Kuleshov and Ermon [1, Section "Bayesian network"].

We illustrate the utility of Proposition 1.2 in the following two examples.

Example 1.3 (Markov chain)**.** Let $V = \{1, \ldots, N\}$ and $\text{pa}(1) = \varnothing,\ \text{pa}(i+1) = \{i\}$ for $1 \leq i < N$. That is to say, $p(X = x) = p(X_1 = x_1) \prod_{i=i}^{N-1} p(X_{i+1} = x_{i+1} | X_i = x_i)$.

For this example, Proposition 1.2 is equivalent to the Markov property, which says that value of random variable X_{i+1} depends on the past $(X_j)_{j=1}^{i}$ only through X_i.

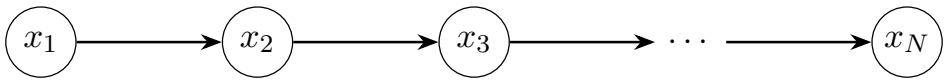

Figure 1: A Markov chain from Example 1.3

Example 1.4 (Probabilistic state-space model)**.** Let $V = \mathcal{X} \cup \mathcal{Y}$, where $|\mathcal{Y}| = |\mathcal{X}| = N$. With a slight abuse of notation, denote the elements of $\mathcal{X}$ by x_i and the elements of $\mathcal{Y}$ by y_i for $1 \leq i \leq N$.

Define the edges in the directed, acyclic graph $\mathcal{G}$ on V by $\text{pa}(x_1) = \varnothing,\ \text{pa}(x_{i+1}) = \{x_i\}$ for $1 \leq i < N$ and $\text{pa}(y_i) = \{x_i\}$ for $1 \leq i \leq N$. Let $(X_i, Y_i)_{i=1}^{N}$ denote the collection of random

variables corresponding to V. Then the likelihood of $(X_i, Y_i)_{i=1}^N$ is

$$p(X = x, Y = y)$$
$$= p(X_1 = x_1)p(Y_1 = y_1|X_1 = x_1) \prod_{i=1}^{N-1} p(Y_{i+1} = y_{i+1}|X_{i+1} = x_{i+1})p(X_{i+1} = x_{i+1}|X_i = x_i).$$

In this BN, Proposition 1.2 tells us that $(X_i)_{i=1}^N$ obeys the Markov property. On the other hand, the random variables $(Y_i)_{i=1}^N$ do *not* obey the Markov property. However, Proposition 1.2 tells us that Y_i and Y_{i+1} are conditionally independent given X_i.

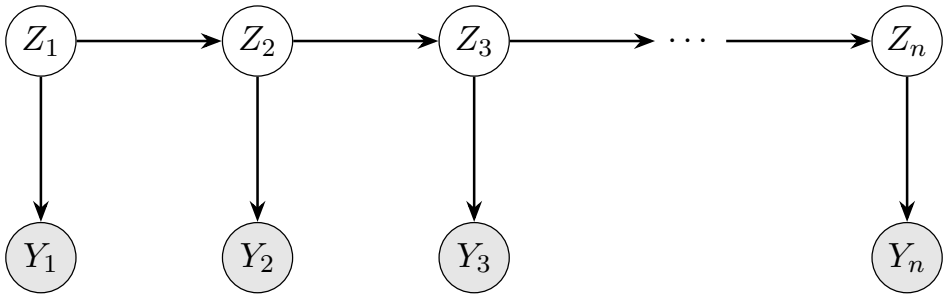

Figure 2: A probabilistic state-space model from Example 1.4

1.2 Inference in probabilistic SSMs

For the remainder of the lecture, we will focus on probabilistic state-space models wherein $\mathcal{Y} \subset \mathbb{R}^d$ and $\mathcal{X} \subset \mathbb{R}^k$. In particular, we will suppose that we receive realizations of random variables $(Y_i)_{i=1}^N$, which we will refer to as the **observations**, but not $(X_i)_{i=1}^N$, which we will refer to as our **latent variables/states**. Our primary goal will be to infer a posterior distribution over the latent variables, given the observations.

In any probabilistic state-space model, the posterior over latent variable is a Markov chain. This is because by Bayes' rule, $p(X = x|Y = y) \propto p(X = x, Y = y)$. Consequently,

$$p(X_i = x_i|Y = y, X_{1:i-1} = x_{1:i-1}) \propto p(X_{1:i} = x_{1:i}, Y = y)$$
$$= \sum_{x_{i+1:N}} p(X_{1:i} = x_{1:i}, X_{i+1:N} = x_{i+1:N}, Y = y)$$
$$\propto p(Y_i = y_i|X_i = x_i)p(X_i = x_i|X_{i-1} = x_{i-1}) \sum_{x_{i+1:N}} \prod_{j=i}^{N} p(X_{i+1} = x_{i+1}|X_i = x_i)p(Y_{i+1} = y_{i+1}|X_{i+1} = x_{i+1}).$$

Observe that the sum is a function of x_i only. This establishes the Markov property of the posterior distribution. We will also briefly discuss parameter estimation in Section 4.

Latent state inference in probabilistic SSMs is relevant to many important applications, including trajectory tracking/navigation, economics/finance, robotics, biotechnology (e.g., wearables), climatology, speech/audio, neuroscience, among others.

2 Hidden Markov Models (HMMs)

A **hidden Markov model** (HMM) is a probabilistic state-space model with latent variables taking on a finite number of values $\mathcal{X} = \{1, \ldots, k\}$.

The **transition distribution** of the latent state is $p(X_{i+1} = x_{i+1}|X_i = x_i) = \text{Cat}(P_{x_i})$, where $P \in \mathbb{R}^{k \times k}$ is a matrix whose rows sum to 1 (i.e., row stochastic), P_{x_i} denotes the x_ith row of

P, and $\text{Cat}(p)$ represents the categorical distribution for each $p \in \mathbb{R}^k,\ p^\top \mathbb{1} = 1$. The transition matrix P may also be chosen to depend on i, in which case $(X_i)_{i=1}^N$ is a time-inhomogeneous Markov chain.

The **emission distribution** is $p(Y_i = y_i | X_i = x_i)$ and is arbitrary; $\mathcal{Y}$ can be either a finite or infinite set. One example for the setting in which $\mathcal{Y} = \mathbb{R}$ is $p(Y_i = y_i | X_i = x_i) = \mathcal{N}(y_i | \mu_{x_i}, \sigma^2)$, where $\mu_1, \ldots, \mu_k \in \mathbb{R}$ are means that corresponding to each value of the latent state.

As discussed in Section 1.2, we will aim to infer a posterior distribution $p(X = x | Y = y)$ over the discrete latent state, given the observations $(Y_i)_{i=1}^N = (y_i)_{i=1}^N$. Since this posterior is a Markov chain, it is enough to determine the **posterior pairwise marginal distributions** $p(X_{i+1} = x_{i+1}, X_i = x_i | Y = y), i = 1, \ldots, N-1$. We do so via the **forward-backward algorithm**.

2.1 Forward-backward algorithm

The forward-backward algorithm involves propagating a sequence of messages from the beginning and end of the time series $(y_i)_{i=1}^N$ that are used to compute the posterior pairwise marginals.

Forward messages

In particular, define the **forward message** at index i by $\alpha_i(x_i) = p(X_i = x_i, Y_{1:i-1} = y_{1:i-1})$ for $1 \le i \le N$ with $\alpha_1(x_1) = p(X_1 = x_1)$. By the definition of the probabilistic SSM, we can rewrite the forward message at index $i+1$ in terms of the forward message at index i:

$$
\begin{aligned}
&\alpha_{i+1}(x_{i+1}) = p(X_{i+1} = x_{i+1}, Y_{1:i} = y_{1:i}) \\
&= \sum_{x_{1:i}} p(X_{i+1} = x_{i+1} | X_i = x_i) p(Y_i = y_i | X_i = x_i) \prod_{j=1}^{i-1} p(X_{j+1} = x_{j+1} | X_j = x_j) p(Y_j = y_j | X_j = x_j) \\
&= \sum_{x_i} p(X_{i+1} = x_{i+1} | X_i = x_i) p(Y_i = y_i | X_i = x_i) \left(\sum_{x_{1:i-1}} \prod_{j=1}^{i-1} p(X_{j+1} = x_{j+1} | X_j = x_j) p(Y_j = y_j | X_j = x_j) \right) \\
&= \sum_{x_i} p(X_{i+1} = x_{i+1} | X_i = x_i) p(Y_i = y_i | X_i = x_i) \alpha_i(x_i). \qquad (5)
\end{aligned}
$$

Equation (5) can be rewritten compactly in matrix notation as $\alpha_{i+1} = P^\top(\alpha_i \odot l_i)$, where $\alpha_i = [\alpha_i(1), \ldots, \alpha_i(k)]^\top \in \mathbb{R}^k$ and $l_i = [p(Y_i = y_i | X_i = 1), \ldots, p(Y_i = y_i | X_i = k)]^\top \in \mathbb{R}^k$. $\odot$ denotes the Hadamard (element-wise) product.

The cost of computing all forward messages from time $i = 0$ to time $i = N$ is $\mathcal{O}(k^2 N)$. Once one has computed the forward messages, they can compute the **posterior predictive distributions** at times $i = 1, \ldots, N-1$

$$p(X_{i+1} = x_{i+1} | Y_{1:i} = y_{1:i}) = \alpha_{i+1}(x_{i+1}) / \alpha_{i+1}^\top \mathbb{1},$$

where $\alpha_{i+1}^\top \mathbb{1}$ is simply the component-wise sum of the vector α_{i+1}.

Backward messages

Similarly, we define the **backward message** at index i by $\beta_i(x_i) = p(Y_{i+1:N} = y_{i+1:N} | X_i = x_i)$ for $1 \le i \le N$ with $\beta_N(x_N) = 1$. However, unlike the forward message, the backward message satisfies a recursion backward in time. Specifically, we can rewrite the backward message at index $i-1$ in terms of the backward message at index i:

$$\beta_{i-1}(x_{i-1}) = p(Y_{i:N} = y_{i:N} | X_{i-1} = x_{i-1})$$

$$= \sum_{x_i} p(X_i = x_i | X_{i-1} = x_{i-1}) p(Y_i = y_i | X_i = x_i) p(Y_{i+1:N} = y_{i+1:N} | X_i = x_i)$$
$$= \sum_{x_i} p(X_i = x_i | X_{i-1} = x_{i-1}) p(Y_i = y_i | X_i = x_i) \beta_i(x_i).$$

In matrix notation, $\beta_{i-1} = P(\beta_i \odot l_i)$, where $l_i \in \mathbb{R}^k$ is defined as previously and $\beta_i = [\beta_i(1), \ldots, \beta_i(k)] \in \mathbb{R}^k$.

The backward messages from index $i = N$ to $i = 1$ can be computed with computational complexity $\mathcal{O}(k^2 N)$.

Putting it all together

Now that we have shown how to compute the forward and backward messages, we are prepared to compute the posterior marginals and pairwise marginals.

First, for the posterior marginals, we have by Bayes' rule and Proposition 1.2

$$\begin{aligned} &p(X_i = x_i | Y = y) \propto p(X_i = x_i, Y = y) \\ &= p(Y_{i+1:N} = y_{i+1:N} | X_i = x_i, Y_{1:i} = y_{1:i}) p(X_i = x_i, Y_{1:i} = y_{1:i}) \\ &= p(Y_{i+1:N} = y_{i+1:N} | X_i = x_i) p(Y_i = y_i | X_i = x_i) p(X_i = x_i, Y_{1:i-1} = y_{1:i-1}) \\ &= \beta_i(x_i) \times l_i(x_i) \times \alpha_i(x_i), \end{aligned}$$

where $l_i(x_i) = p(Y_i = y_i | X_i = x_i)$. Hence,

$$p(X_i = x_i | Y = y) = \beta_i(x_i) l_i(x_i) \alpha_i(x_i) / (\beta_i \odot l_i \odot \alpha_i)^\top \mathbb{1}$$

yields the posterior marginals.

For the posterior pairwise marginals, a similar calculation gives

$$\begin{aligned} &p(X_{i+1} = x_{i+1}, X_i = x_i | Y = y) \propto p(X_{i+1} = x_{i+1}, X_i = x_i, Y = y) \\ &= p(Y_{i+2:N} = y_{i+2:N} | X_{i+1} = x_{i+1}) p(Y_{i+1} = y_{i+1} | X_{i+1} = x_{i+1}) \times \\ &p(X_{i+1} = x_{i+1} | X_i = x_i) p(Y_i = y_i | X_i = x_i) p(X_i = x_i, Y_{1:i-1} = y_{1:i-1}) \\ &= \beta(x_{i+1}) \times l_{i+1}(x_{i+1}) \times p(X_{i+1} = x_{i+1} | X_i = x_i) \times l_i(x_i) \times \alpha(x_i) \end{aligned}$$

Hence,

$$p(X_i = x_i, X_{i+1} = x_{i+1} | Y = y) = \frac{\beta(x_{i+1}) l_{i+1}(x_{i+1}) p(X_{i+1} = x_{i+1} | X_i = x_i) l_i(x_i) \alpha(x_i)}{\sum_{x_i, x_{i+1}} \beta(x_{i+1}) l_{i+1}(x_{i+1}) p(X_{i+1} = x_{i+1} | X_i = x_i) l_i(x_i) \alpha(x_i)}$$

yields the posterior pairwise marginals.

Remark 2.1. While the posterior distribution over latent states $(X_i)_{i=1}^N$ given observations $(Y_i)_{i=1}^N$ is a Markov chain, it is *not* a time-homogeneous Markov chain, even when the Markov chain that defines the prior distribution $p(X = x)$ is time-homogeneous.

Altogether, we have shown that by propagating a sequence of the forward and backward messages via the forward-backward algorithm, we can compute the posterior distribution over the latent state $(X_i)_{i=1}^N$, which is a Markov chain, given observations $(Y_i)_{i=1}^N$. Moreover, these messages can be computed efficiently: with computational complexity quadratic in the number of states k and linear in the length of the time series N.

3 Linear Gaussian State-space Models

The forward-backward algorithm presented on Section 2 appears quite general, and it only relies on the assumption that $\mathcal{X}$ is finite in order to normalize the forward and backward messages. This algorithm is a specific instance of a **message passing algorithm** for probabilistic graphical models, applied to a probabilistic state-space model.

To derive an analogous algorithm for a probabilistic state-space model with continuous latent state $\mathcal{X} \subseteq \mathbb{R}^k$, one must simply replace the summations with integrals. That is to say,

$$\begin{aligned}\alpha_{i+1}(x_{i+1}) &= \int p(X_{i+1} = x_{i+1}|X_i = x_i)p(Y_i = y_i|X_i = x_i)\alpha_i(x_i)dx_i, \quad \alpha_1(x_1) = p(X_1 = x_1) \\ \beta_{i-1}(x_{i-1}) &= \int p(X_i = x_i|X_{i-1} = x_{i-1})p(Y_i = y_i|X_i = x_i)\beta_i(x_i)dx_i, \quad \beta_N(x_N) \propto 1\end{aligned} \tag{6}$$

Equation (6) calls to mind numerous practical complications to this algorithm. First, for arbitrary transition distribution $p(X_{i+1} = x_{i+1}|X_i = x_i)$ and emission distribution $p(Y_i = y_i|X_i = x_i)$, the forward and backward messages are defined by an intractable integral. And so while the forward-backward algorithm is theoretically sound for arbitrary transition and emission distributions, it is not possible to implement.

Second, there is no uniform probability distribution over $\mathbb{R}^k$, and so the final backward message $\beta_N(x_N) \propto 1$ is ill-defined. One solution is to define the backward messages only for times $1 \leq i < N$. The final posterior marginal can be calculated from $\alpha_N(x_N)$ and l_N only:

$$p(X_N = x_N|Y = y) \propto p(X_N = x_N, Y_{1:N-1} = y_{1:N-1})p(Y_N = y_N|X_N = x_N) = \alpha_N(x_N)l_N(x_N).$$

Lastly, related to the first point, the forward and backward messages must be normalized to yield the posterior marginals and pairwise marginals.

Model overview

Throughout this section, we will focus our attention on a setting where $\mathcal{X} = \mathbb{R}^k$, $\mathcal{Y} = \mathbb{R}^d$ but the forward and backward messages are tractable. Indeed, we will study posterior inference in the **linear Gaussian state space model** (LGSSM). This model is specified by

- initial distribution $p(X_0 = x_0) = \mathcal{N}(x_0|\nu, V)$, where $\nu \in \mathbb{R}^k$ and $V \in \mathbb{R}^{k\times k}$ positive definite
- transition distribution $p(X_{i+1} = x_{i+1}|X_i = x_i) = \mathcal{N}(x_{i+1}|Ax_i + b, Q)$, where $A \in \mathbb{R}^{k\times k}$, $b \in \mathbb{R}^k$, and $Q \in \mathbb{R}^{k\times k}$ positive definite
- emission distribution $p(Y_i = y_i|X_i = x_i) = \mathcal{N}(y_i|Cx_i + d, R)$, where $C \in \mathbb{R}^{d\times k}$, $d \in \mathbb{R}^d$, and $R \in \mathbb{R}^{d\times d}$ positive definite

The reason for focusing on the linear Gaussian state space model is that the multivariate Gaussian distribution satisfies nice conditioning properties. We recall these properties in the following proposition:

Proposition 3.1 (Gaussian conditioning)**.** *Let $A \in \mathbb{R}^{k\times k}$, $b \in \mathbb{R}^k$, and let $R \in \mathbb{R}^{k\times k}$ be symmetric, positive definite. Moreover, let $m_y, m_z \in \mathbb{R}^k$ and let $S_y, S_z \in \mathbb{R}^{k\times k}$ be symmetric, positive definite.*

(i) Suppose random variables $Y, Z \in \mathbb{R}^k$ are defined such that $Z|Y \overset{d}{=} \mathcal{N}(z|Ay + b, R)$ and $Y \overset{d}{=} \mathcal{N}(y|m_y, S_y)$. Then $Z \overset{d}{=} N(z|Am_y + b, AS_yA^\top + R)$.

(ii) Suppose

$$\begin{pmatrix} Y \\ Z \end{pmatrix} \overset{d}{=} \mathcal{N}\left(\begin{pmatrix} y \\ z \end{pmatrix} \middle| \begin{pmatrix} m_y \\ m_z \end{pmatrix}, \begin{pmatrix} S_y & S_{zy}^\top \\ S_{zy} & S_{zz} \end{pmatrix} \right).$$

Then the conditional distribution of Z given Y is

$$Z|(Y=y) \overset{d}{=} \mathcal{N}(z| \underbrace{m_z + S_{zy}S_y^{-1}(y-m_y)}_{m_{z|y}}, \underbrace{S_z - S_{zy}S_y^{-1}S_{zy}^\top}_{S_{z|y}}).$$

*The matrix $S_{z|y}$ is called the **Shur complement** of S_y in the full covariance matrix.*

The proof of Proposition 3.1 is left to the reader and is good practice working with the definition of multivariate normality.

Kalman filtering

The **Kalman filter** is an algorithm for computing the *normalized* forward messages in an LGSSM i.e., the posterior predictive distributions $p(X_i = x_i|Y_{1:i-1} = y_{1:i-1})$. Like computing the forward messages in an HMM (Section 2), the algorithm is recursive.

Suppose we have $p(X_i = x_i|Y_{1:i-1} = y_{1:i-1}) = \mathcal{N}(x_i|m_i, S_i)$ for $m_i \in \mathbb{R}^k$, $S_i \in \mathbb{R}^{k\times k}$ symmetric, positive definite. Then by marginalizing over x_i and invoking the Markov property,

$$p(X_{i+1} = x_{i+1}|Y_{1:i} = y_{1:i}) = \int p(X_{i+1} = x_{i+1}|X_i = x_i)p(X_i = x_i|Y_{1:i} = y_{1:i})dx_i \tag{7}$$

The Kalman filter is comprised of two steps: (i) computing $p(X_i = x_i|Y_{1:i} = y_{1:i})$ (the **update step**) and (ii) integrating equation (7) over x_i (the **predict step**).

Update step. The update step relies on both conditioning properties described in Proposition 3.1. In particular, by property (i) we have

$$\begin{aligned} p(Y_i = y_i|Y_{1:i-1} = y_{1:i-1}) &= \int p(Y_i = y_i|X_i = x_i)p(X_i = x_i|Y_{1:i-1} = y_{1:i-1})dx_i \\ &= \mathcal{N}(y_i|Cm_i + d, CS_iC^\top + R) \\ m_{i,y} = Cm_i + d, &\quad S_{i,y} = CS_iC^\top + R \end{aligned}$$

We can also compute the covariance between X_i and Y_i, conditional on $Y_{1:i-1} = y_{1:i-1}$

$$\mathrm{Cov}(X_i, Y_i|Y_{1:i-1}) = \mathrm{Cov}(X_i, CX_i + d|Y_{1:i-1}) = S_iC^\top.$$

Combining these results, the joint distribution of (X_i, Y_i), conditional on $Y_{1:i-1} = y_{1:i-1}$, is

$$\mathcal{N}\left(\begin{pmatrix} x_i \\ y_i \end{pmatrix} \middle| \begin{pmatrix} m_i \\ m_{i,y} \end{pmatrix}, \begin{pmatrix} S_i & S_iC^\top \\ CS_i & S_{i,y} \end{pmatrix} \right).$$

By Proposition 3.1, property (ii) we deduce

$$p(X_i = x_i|Y_{1:i} = y_{1:i}) = \mathcal{N}(x_i|m_i + S_iC^\top S_{i,y}^{-1}(y_i - m_{i,y}), S_i - S_iC^\top S_{i,y}^{-1}CS_i)$$

We can equivalently write the mean and covariance as

$$\begin{aligned} \tilde{m}_i &= m_i + K_i\tilde{y}_i, \quad \tilde{S}_i = (\mathbb{I}_k - K_iC)S_i \\ K_i &= S_iC^\top(CS_iC^\top + R)^{-1}, \quad \tilde{y}_i = y_i - (Cm_i + d) \end{aligned}$$

K_i is known as the **Kalman gain matrix** and $\tilde{y}_i$ is known as the **innovation**.

Predict step. Once we have computed the distribution of X_i, conditional on $Y_{1:i} = y_{1:i}$, $p(X_i = x_i | Y_{1:i} = y_{1:i})$, it only remains to evaluate the integral equation (7).

Since $p(X_{i+1} = x_{i+1} | X_i = x_i) = \mathcal{N}(x_{i+1} | Ax_i + b, Q)$, property (i) from Proposition 3.1 gives

$$p(X_{i+1} = x_{i+1} | Y_{1:i} = y_{1:i}) = \mathcal{N}(x_{i+1} | m_{i+1}, S_{i+1})$$
$$m_{i+1} = A\tilde{m}_i + b, \quad S_{i+1} = A\tilde{S}_i A^\top + Q.$$

Running the Kalman filter from $i = 1$ to $i = N$ incurs computational cost $\mathcal{O}(N(k^3 + d^3))$.

Rauch-Tung-Striebel smoothing

The Kalman filter computes the predictive distributions $p(X_{i+1} = x_{i+1} | Y_{1:i} = y_{1:i})$, or equivalently the normalized forward messages $\alpha_{i+1}(x_{i+1})$. It remains to show how these distributions can be used to compute the posterior marginals and pairwise marginals.

Rather than computing the posterior distribution by propagating a sequence of backward messages, as is done in Section 2, we describe an alternative approach, the **Rauch-Tung-Striebel smoother** (RTS smoother).

To derive the RTS smoother, we first observe that by Proposition 1.2, for each $1 \leq i < N$,

$$\begin{aligned} &p(X_i = x_i, X_{i+1} = x_{i+1} | Y = y) = p(X_i = x_i | X_{i+1} = x_{i+1}, Y = y) p(X_{i+1} = x_{i+1} | Y = y) \\ &= p(X_i = x_i | X_{i+1} = x_{i+1}, Y_{1:i} = y_{1:i}) p(X_{i+1} = x_{i+1} | Y = y). \end{aligned} \tag{8}$$

The posterior pairwise marginals can be computed recursively, starting from $p(X_N = x_N | Y = y)$, which is computed from the terminal step of the Kalman filter

$$p(X_N = x_N | Y = y) = \mathcal{N}(x_N | m_N^s, S_N^s), \quad m_N^s = m_N + K_N \tilde{y}_N, \quad S_N^s = (\mathbb{I}_k - K_N C) S_N.$$

Let $p(X_{i+1} = x_{i+1} | Y = y) = \mathcal{N}(x_{i+1} | m_{i+1}^s, S_{i+1}^s)$. Observe that from the predict step of the Kalman filter, we know the (joint) distribution of (X_i, X_{i+1}), conditional on $Y_{1:i} = y_{1:i}$

$$\mathcal{N}\left(\begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix}, \begin{pmatrix} \tilde{m}_i \\ m_{i+1} \end{pmatrix}, \begin{pmatrix} \tilde{S}_i & \tilde{S}_i A^\top \\ A\tilde{S}_i & S_{i+1} \end{pmatrix} \right).$$

By Proposition 3.1, property (ii), this allows us to compute

$$p(X_i = x_i | X_{i+1} = x_{i+1}, Y_{1:i} = y_{1:i}) = \mathcal{N}(x_i | \overline{m}_i, \overline{S}_i)$$
$$\overline{m}_i = \tilde{m}_i + \tilde{S}_i A^\top S_{i+1}^{-1}(x_{i+1} - m_{i+1}) \quad \overline{S}_i = \tilde{S}_i - \tilde{S}_i A^\top S_{i+1}^{-1} A \tilde{S}_i.$$

Finally we combine this with equation (8) using Proposition 3.1, property (i) to obtain the pairwise posterior distribution of (X_i, X_{i+1}), conditional on $Y = y$:

$$\mathcal{N}\left(\begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix}, \begin{pmatrix} m_i^s \\ m_{i+1}^s \end{pmatrix}, \begin{pmatrix} S_i^s & S_{i,i+1}^s \\ (S_{i,i+1}^s)^\top & S_{i+1}^s \end{pmatrix} \right),$$
$$m_i^s = \tilde{m}_i + \tilde{S}_i A^\top S_{i+1}^{-1}(m_{i+1}^s - m_{i+1}), \quad S_i^s = \tilde{S}_i + \tilde{S}_i A^\top S_{i+1}^{-1}(S_{i+1}^s - S_{i+1}) S_{i+1}^{-1} A \tilde{S}_i,$$
$$S_{i,i+1}^s = \tilde{S}_i A^\top S_{i+1}^{-1} S_{i+1}^s.$$

To summarize, we have shown that for a state-space model in which both the transition and emission distributions are linear and Gaussian, exact posterior inference for the latent states $(X_i)_{i=1}^N$, conditional on the observations $(Y_i)_{i=1}^N = (y_i)_{i=1}^N$, is tractable via the Kalman filter and Rauch-Tung-Striebel smoother. These algorithms rely on the properties of the multivariate normal distribution (Proposition 3.1). In the case that the transition or emission distribution is Gaussian and nonlinear or non-Gaussian, exact posterior inference is intractable. In this case, one needs to resort to approximate inference algorithms (e.g., MCMC, particle filtering, variational inference).

4 Parameter estimation

So far, we have discussed the problem of posterior inference in a probabilistic state-space model for the latent variables $(X_i)_{i=1}^N$, given the observation $(Y_i)_{i=1}^N = (y_i)_{i=1}^N$. In this section, we discuss a different but related problem, parameter estimation.

Consider learning the parameters θ of the transition distribution $p_\theta(X_{i+1} = x_{i+1}|X_i = x_i)$ and/or of the emission distribution $p_\theta(Y_i = y_i|X_i = x_i)$ in a probabilistic state-space model. For example, in a HMM, θ would be the transition matrix P as well as the parameters of the emission distribution. In a LGSSM with fixed noise covariance R, Q, the parameters would be $\theta = \{A, b, C, d\}$.

If we had observed both $(Y_i)_{i=1}^N$ and $(X_i)_{i=1}^N$, then we could write out the log-likelihood via equation (1) and perform maximum likelihood estimation. However, if we observe only $(Y_i)_{i=1}^N$, writing down the marginal log-likelihood of $(Y_i)_{i=1}^N$ is not immediate.

To see how we can compute the marginal log-likelihood, observe

$$\begin{aligned}\log p_\theta(Y = y) &\overset{(\star)}{=} \mathbb{E}_{p_\theta(x|y)}[\log p_\theta(Y = y)] \\ &\overset{(\star\star)}{=} \mathbb{E}_{p_\theta(x|y)}[\log p_\theta(X = x) + \log p_\theta(Y = y|X = x) - \log p_\theta(X = x|Y = y)] \\ &= \mathbb{E}_{p_\theta(x|y)}[\log p_\theta(Y = y|X = x)] + \mathbb{E}_{p_\theta(x|y)}\left[\log\left\{\frac{p_\theta(X = x)}{p_\theta(X = x|Y = y)}\right\}\right] \qquad (9)\end{aligned}$$

Equality $(\star)$ holds because $\log p_\theta(Y = y)$ is a constant with respect to x. And equality $(\star\star)$ holds by adding and subtracting $\mathbb{E}_{p_\theta(x|y)}[\log p_\theta(X = x, Y = y)]$ and then applying Bayes' rule. The second term on the right-hand side of equation (9) is equal to the negative **KL divergence** between $p_\theta(X = x|Y = y)$ and $p_\theta(X = x)$.

Equation (9) gives us a strategy for computing the marginal log-likelihood of $(Y_i)_{i=1}^N$ in a probabilistic state-space model. First, run the forward-backward algorithm to compute the posterior distribution $p_\theta(X = x|Y = y)$. Then, use the posterior distribution to compute the right-hand side of equation (9).

To perform maximum-likelihood estimation, one can differentiate (9) with respect to the transition and/or likelihood parameters and then update the parameters via gradient ascent:

$$\theta^{(t+1)} \leftarrow \theta^{(t)} + \eta\nabla_\theta\left(\mathbb{E}_{p_\theta(x|y)}[\log p_\theta(Y = y|X = x)] + \mathbb{E}_{p_\theta(x|y)}\left[\log\left\{\frac{p_\theta(X = x)}{p_\theta(X = x|Y = y)}\right\}\right]\right)\Big|_{\theta=\theta^{(t)}}.$$

Note that the dependence on the parameters is not only in the integrand of (9), but also in the distribution $p_\theta(X = x|Y = y)$ with respect to which the expectation is taken. At the next iteration of gradient ascent, one will need to recompute the posterior distribution $p_\theta(X = x|Y = y)$ at the new set of parameters $\theta^{(t+1)}$.

Expectation-maximization algorithm

The **expectation-maximization** algorithm (EM) is an alternative to the maximum likelihood algorithm that we sketched. Namely, whereas maximum likelihood explicitly accounts for the dependence of the expectation in equation (9) on the model parameters θ, EM ignores this dependence.

In the EM algorithm, the posterior $p_\theta(X = x|Y = y)$ is first computed at the current set of model parameters $\theta = \theta^{(t)}$. This step is called the **E-step**. One then updates θ either by solving

$$\mathbb{E}_{p_{\theta^{(t)}}(x|y)}\left[\nabla_\theta\left(\log p(Y = y|X = x) + \log\left\{\frac{p(X = x)}{p(X = x|Y = y)}\right\}\right)\right] = 0$$

analytically or via gradient ascent. This is called the **M-step**. EM has the advantage that the M-step updates often exist in closed-form, whereas those for maximum likelihood do not.

The EM algorithm is described in pseudocode in Algorithm 1.

Algorithm 1 Expectation-Maximization (EM) for probabilistic state-space models

Require: Observations $(y_i)_{i=1}^N$, initial parameters $\theta^{(0)}$, number of iterations T

1: **for** $t = 0, 1, \ldots, T-1$ **do**

2: **E-step:** Compute the smoothing distribution under $\theta^{(t)}$:

$$p_{\theta^{(t)}}(X_{1:N} = x_{1:N} | Y_{1:N} = y_{1:N})$$

3: **M-step:** Update parameters by maximizing the expected complete-data log-likelihood:

$$\theta^{(t+1)} \in \arg\max_{\theta} \; Q(\theta\,; \theta^{(t)}), \qquad Q(\theta\,; \theta^{(t)}) := \mathbb{E}_{p_{\theta^{(t)}}(x_{1:N}|y_{1:N})}[\log p_\theta(x_{1:N}, y_{1:N})]\,.$$

4: **end for**

5: **return** $\theta^{(T)}$

References

[1] V. Kuleshov and S. Ermon. Probabilistic graphical models. https://ermongroup.github.io/cs228-notes/, 2025. Lecture notes for CS 228.

[2] S. L. Linderman. Models and algorithms for discrete data. https://slinderman.github.io/stats305b/, 2025. Lecture notes for STATS 305B.