CHAPTER 24

Continuous-state Markov Models

Many physical systems can be understood as continuous variable models undergoing a transition dependent only the state of the system at the previous time. In this chapter we discuss some of the classical models in this area which are highly specialised to retain tractability of inference. These models are found in a wide variety of fields from finance to signal processing in engineering.

24.1 Observed Linear Dynamical Systems

In many practical timeseries applications the data is naturally continuous, particularly for models of the physical environment. In contrast to discrete-state Markov models, chapter(23), continuous state distributions are not automatically closed under operations such as products and marginalisation. To make practical algorithms for which inference and learning can be carried efficiently, we therefore are heavily restricted in the form of the continuous transition $p(v_t|v_{t-1})$. A simple yet powerful class of such transitions are the linear dynamical systems. A deterministic observed linear dynamical system[1] (OLDS) defines the temporal evolution of a vector $\mathbf{v}_t$ according to the discrete-time update equation

$$\mathbf{v}_t = \mathbf{A}_t \mathbf{v}_{t-1} \tag{24.1.1}$$

where $\mathbf{A}_t$ is the transition matrix at time t. For the case that $\mathbf{A}_t$ is invariant with t, the process is called time-invariant, which we assume throughout unless explicitly stated otherwise.

A motivation for studying OLDSs is that many equations that describe the physical world can be written as an OLDS. OLDSs are interesting since they may be used as simple prediction models: if $\mathbf{v}_t$ describes the state of the environment at time t, then $\mathbf{A}\mathbf{v}_t$ predicts the environment at time $t+1$. As such, these models, have widespread application in many branches of science, from engineering and physics to economics.

The OLDS equation (24.1.1) is deterministic so that if we specify $\mathbf{v}_1$, all future values $\mathbf{v}_2, \mathbf{v}_3, \ldots$, are defined. For a $\dim \mathbf{v} = V$ dimensional vector, its evolution is described by (assuming $\mathbf{A}$ is diagonalisable)

$$\mathbf{v}_t = \mathbf{A}^{t-1}\mathbf{v}_1 = \mathbf{P}\mathbf{\Lambda}^{t-1}\mathbf{P}^{-1}\mathbf{v}_1 \tag{24.1.2}$$

where $\mathbf{\Lambda} = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_V)$, is the diagonal eigenvalue matrix, and $\mathbf{P}$ is the corresponding eigenvector matrix of $\mathbf{A}$. If $\lambda_i > 1$ then for large t, $\mathbf{v}_t$ will explode. On the other hand, if $\lambda_i < 1$, then λ_i^{t-1} will tend to zero. For stable systems we require therefore no eigenvalues of magnitude greater than 1 and only unit eigenvalues will contribute in long term. Note that the eigenvalues may be complex which corresponds to rotational behaviour, see exercise(24.1). More generally, we may consider additive noise on $\mathbf{v}$ and define a stochastic OLDS.

[1] We use the terminology 'observed' LDS to differentiate from the more general LDS state-space model. In some texts, however, the term LDS is applied to the models under discussion in this chapter.

Definition 24.1 (Observed Linear Dynamical System)**.**

$$\mathbf{v}_t = \mathbf{A}_t \mathbf{v}_{t-1} + \boldsymbol{\eta}_t \tag{24.1.3}$$

where $\boldsymbol{\eta}_t$ is a noise vector sampled from a Gaussian distribution,

$$\mathcal{N}(\boldsymbol{\eta}_t | \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \tag{24.1.4}$$

This is equivalent to a first order Markov model

$$p(\mathbf{v}_t | \mathbf{v}_{t-1}) = \mathcal{N}(\mathbf{v}_t | \mathbf{A}_t \mathbf{v}_{t-1} + \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \tag{24.1.5}$$

At $t = 1$ we have an initial distribution $p(\mathbf{v}_1) = \mathcal{N}(\mathbf{v}_1 | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$. For $t > 1$ if the parameters are time-independent, $\boldsymbol{\mu}_t \equiv \boldsymbol{\mu}$, $\mathbf{A}_t \equiv \mathbf{A}$, $\boldsymbol{\Sigma}_t \equiv \boldsymbol{\Sigma}$, the process is called time-invariant.

24.1.1 Stationary distribution with noise

Consider the one-dimensional linear system with independent additive noise

$$v_t = a v_{t-1} + \eta_t, \qquad \eta_t \sim \mathcal{N}(\eta_t | 0, \sigma_v^2) \tag{24.1.6}$$

If we start at some state v_1, and then for $t > 1$ recursively sample according to $v_t = a v_{t-1} + \eta_t$, does the distribution of the $v_t, t \gg 1$ tend to a steady, fixed distribution? Assuming that we can represent the distribution of v_{t-1} as a Gaussian with mean μ_{t-1} and variance σ_{t-1}^2, $v_{t-1} \sim \mathcal{N}(v_{t-1} | \mu_{t-1}, \sigma_{t-1}^2)$, then using $\langle \eta_t \rangle = 0$ we have

$$\langle v_t \rangle = a \langle v_{t-1} \rangle + \langle \eta_t \rangle \Rightarrow \mu_t = a \mu_{t-1} \tag{24.1.7}$$

$$\left\langle v_t^2 \right\rangle = \langle a v_{t-1} + \eta_t \rangle^2 = a^2 \left\langle v_{t-1}^2 \right\rangle + 2a \langle v_{t-1} \rangle \langle \eta_t \rangle + \left\langle \eta_t^2 \right\rangle \tag{24.1.8}$$

$$\Rightarrow \sigma_t^2 = a^2 \sigma_{t-1}^2 + \sigma_v^2 \tag{24.1.9}$$

so that

$$v_t \sim \mathcal{N}(v_t | a \mu_{t-1}, a^2 \sigma_{t-1}^2 + \sigma_v^2) \tag{24.1.10}$$

Assuming there is a fixed variance σ_∞^2 for the infinite time case, the stationary distribution satisfies

$$\sigma_\infty^2 = a^2 \sigma_\infty^2 + \sigma_v^2 \quad \Rightarrow \sigma_\infty^2 = \frac{\sigma_v^2}{1 - a^2} \tag{24.1.11}$$

Similarly, the mean is given by $\mu_\infty = a^\infty \mu_1$. If $a \geq 1$ the variance increases indefinitely with t. For $a < 1$, the mean tends to zero yet the variance remains finite. Even though the magnitude of v_{t-1} is decreased by a factor of a at each iteration, the additive noise on average boosts the magnitude so that it remains steady in the long run. More generally for a system updating a vector $\mathbf{v}_t$ according to

$$\mathbf{v}_t = \mathbf{A} \mathbf{v}_{t-1} + \boldsymbol{\eta}_t \tag{24.1.12}$$

for the existence of a steady state we require that all eigenvalues of $\mathbf{A}$ must be ≤ 1.

24.2 Auto-Regressive Models

A scalar time-invariant auto-regressive model is defined by

$$v_t = \sum_{l=1}^{L} a_l v_{t-l} + \eta_t, \qquad \eta_t \sim \mathcal{N}(\eta_t | \mu, \sigma^2) \tag{24.2.1}$$

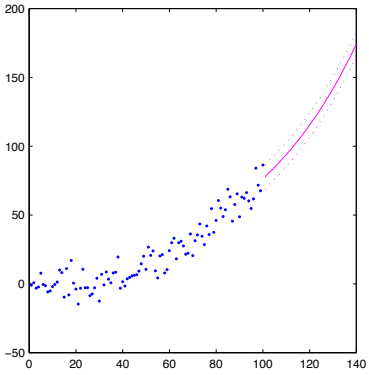

Figure 24.1: Fitting an order 3 AR model to the training points. The x axis represents time, and the y axis the value of the timeseries. The solid line is the mean prediction and the dashed lines $\pm$ one standard deviation. See `demoARtrain.m`

where $\mathbf{a} = (a_1, \ldots, a_L)^\mathsf{T}$ are called the AR coefficients and σ^2 is called the *innovation noise*. The model predicts the future based on a linear combination of the previous L observations. As a belief network, the AR model can be written as an L^{th} order Markov model:

$$p(v_{1:T}) = \prod_{t=1}^{T} p(v_t|v_{t-1}, \ldots, v_{t-L}), \qquad \text{with } v_i = \emptyset \text{ for } i \leq 0 \tag{24.2.2}$$

with

$$p(v_t|v_{t-1}, \ldots, v_{t-L}) = \mathcal{N}\left(v_t \middle| \sum_{l=1}^{L} a_l v_{t-l}, \sigma^2\right) \tag{24.2.3}$$

Introducing the vector of the L previous observations

$$\hat{\mathbf{v}}_{t-1} \equiv [v_{t-1}, v_{t-2}, \ldots, v_{t-L}]^\mathsf{T} \tag{24.2.4}$$

we can write more compactly

$$p(v_t|v_{t-1}, \ldots, v_{t-L}) = \mathcal{N}\left(v_t|\mathbf{a}^\mathsf{T}\hat{\mathbf{v}}_{t-1}, \sigma^2\right) \tag{24.2.5}$$

AR models are heavily used in financial time-series prediction (see for example [274]), being able to capture simple trends in the data. Another common application area is in speech processing whereby for a one-dimensional speech signal partitioned into windows of length T, the AR coefficients best able to describe the signal in each window are found[217]. These AR coefficients then form a compressed representation of the signal and subsequently transmitted for each window, rather than the original signal itself. The signal can then be approximately reconstructed based on the AR coefficients. Such a representation is used for example in telephones and known as a linear predictive vocoder[258].

24.2.1 Training an AR model

Maximum Likelihood training of the AR coefficients is straightforward based on

$$\log p(v_{1:T}) = \sum_{t=1}^{T} \log p(v_t|\hat{\mathbf{v}}_{t-1}) = -\frac{1}{2\sigma^2}\sum_{t=1}^{T}\left(v_t - \hat{\mathbf{v}}_{t-1}^\mathsf{T}\mathbf{a}\right)^2 - \frac{T}{2}\log(2\pi\sigma^2) \tag{24.2.6}$$

Differentiating *w.r.t.* $\mathbf{a}$ and equating to zero we arrive at

$$\sum_t \left(v_t - \hat{\mathbf{v}}_{t-1}^\mathsf{T}\mathbf{a}\right)\hat{\mathbf{v}}_{t-1} = 0 \tag{24.2.7}$$

so that optimally

$$\mathbf{a} = \left(\sum_t \hat{\mathbf{v}}_{t-1}\hat{\mathbf{v}}_{t-1}^\mathsf{T}\right)^{-1}\sum_t v_t\hat{\mathbf{v}}_{t-1} \tag{24.2.8}$$

These equations can be solved by Gaussian elimination. Similarly, optimally,

$$\sigma^2 = \frac{1}{T}\sum_{t=1}^{T}\left(v_t - \hat{\mathbf{v}}_{t-1}^\mathsf{T}\mathbf{a}\right)^2 \tag{24.2.9}$$

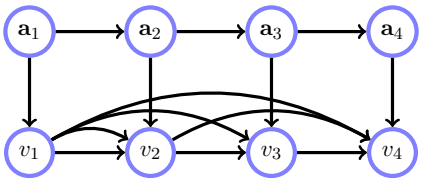

Figure 24.2: A time-varying AR model as a latent LDS. Since the observations are known, this model is a time-varying latent LDS, for which smoothed inference determines the time-varying AR coefficients.

Above we assume that 'negative' timesteps are available in order to keep the notation simple. If times before the window over which we learn the coefficients are not available, a minor adjustment is required to start the summations from $t = L + 1$.

Given a trained $\mathbf{a}$, future predictions can be made using $v_{t+1} = \hat{\mathbf{v}}_t^\mathsf{T}\mathbf{a}$. As we see, the model is capable of capturing the trend in the data.

24.2.2 AR model as an OLDS

We can write equation (24.2.1) as an OLDS using

$$\begin{pmatrix} v_t \\ v_{t-1} \\ \vdots \\ v_{t-L+1} \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & \dots & a_L \\ 1 & 0 & \dots & 0 \\ \vdots & 1 & \dots & 0 \\ 0 & \dots & 1 & 0 \end{pmatrix} \begin{pmatrix} v_{t-1} \\ v_{t-2} \\ \vdots \\ v_{t-L} \end{pmatrix} + \begin{pmatrix} \eta_t \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{24.2.10}$$

We can write equation (24.2.1) as the OLDS

$$\hat{\mathbf{v}}_t = \mathbf{A}\hat{\mathbf{v}}_{t-1} + \boldsymbol{\eta}_t, \qquad \boldsymbol{\eta}_t \sim \mathcal{N}(\boldsymbol{\eta}_t|\mathbf{0}, \boldsymbol{\Sigma}) \tag{24.2.11}$$

where we define the block matrices

$$\mathbf{A} = \left(\begin{array}{c|c} a_{1:L-1} & a_L \\ \hline \mathbf{I} & \mathbf{0} \end{array}\right), \qquad \boldsymbol{\Sigma} = \left(\begin{array}{c|c} \sigma^2 & 0_{1,1:L-1} \\ \hline 0_{1:L-1,1} & 0_{1:L-1,1:L-1} \end{array}\right) \tag{24.2.12}$$

In this representation, the first component of the vector is updated according to the standard AR model, with the remaining components being copies of the previous values.

24.2.3 Time-varying AR model

An alternative to Maximum Likelihood is to view learning the AR coefficients as a problem in inference in a latent LDS, a model which is discussed in detail in section(24.3). If $\mathbf{a}_t$ are the latent AR coefficients, the term

$$v_t = \hat{\mathbf{v}}_{t-1}^\mathsf{T}\mathbf{a}_t + \eta_t, \qquad \eta_t \sim \mathcal{N}(\eta_t|0, \sigma^2) \tag{24.2.13}$$

can be viewed as the emission distribution of a latent LDS in which the hidden variable is $\mathbf{a}_t$ and the time-dependent emission matrix is given by $\hat{\mathbf{v}}_{t-1}^\mathsf{T}$. By placing a simple latent transition

$$\mathbf{a}_t = \mathbf{a}_{t-1} + \boldsymbol{\eta}_t^a, \qquad \boldsymbol{\eta}_t^a \sim \mathcal{N}(\eta_t^a|0, \sigma_a^2\mathbf{I}) \tag{24.2.14}$$

we encourage the AR coefficients to change slowly with time. This defines a model

$$p(v_{1:T}, \mathbf{a}_{1:T}) = \prod_t p(v_t|\mathbf{a}_t, \hat{\mathbf{v}}_{t-1})p(\mathbf{a}_t|\mathbf{a}_{t-1}) \tag{24.2.15}$$

Our interest is then in the conditional $p(\mathbf{a}_{1:T}|v_{1:T})$ from which we can compute the a-posteriori most likely sequence of AR coefficients. Standard smoothing algorithms can then be applied to yield the time-varying AR coefficients, see `demoARlds.m`.

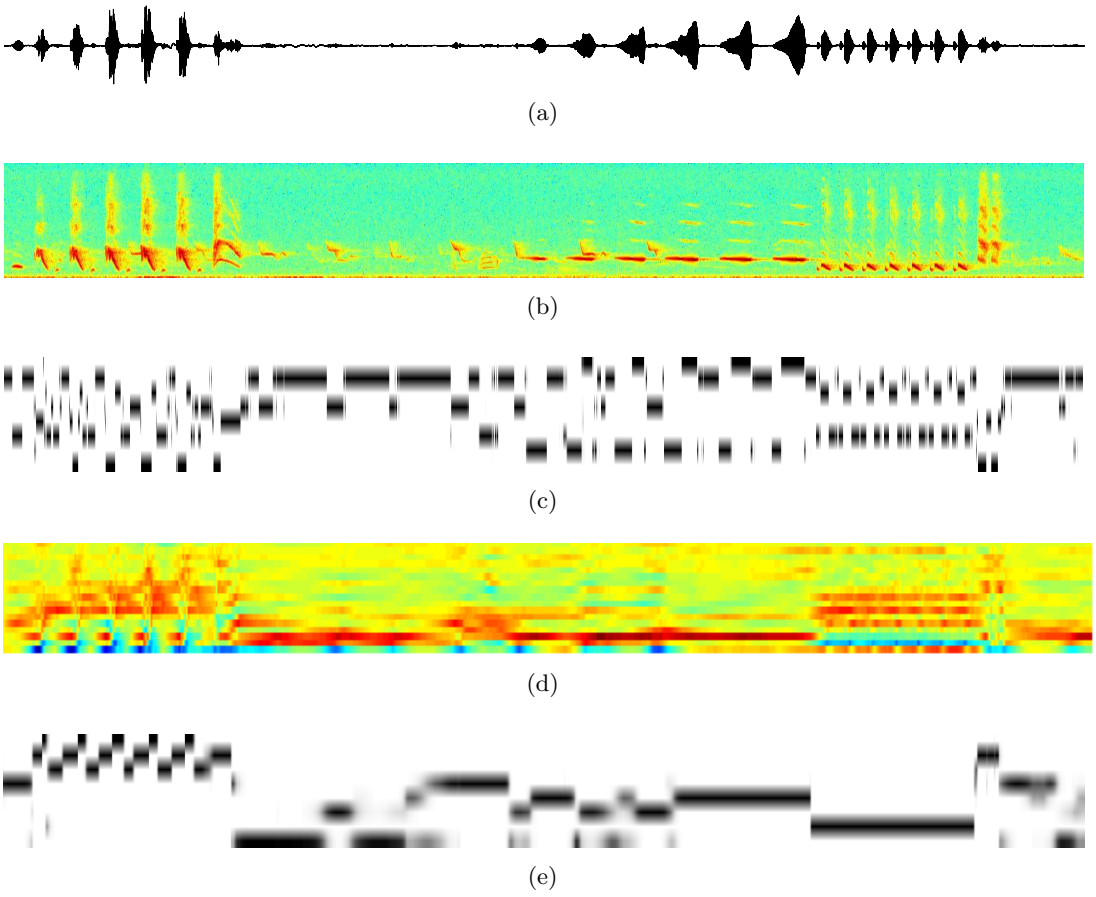

Figure 24.3: **(a)**: The raw recording of 5 seconds of a nightingale song (with additional background birdsong). **(b)**: Spectrogram of (a) up to 20,000 Hz. **(c)**: Clustering of the results in panel (b) using an 8 component Gaussian mixture model. The index (from 1 to 8) of the component most probably responsible for the observation is indicated vertically in black. **(d)**: The 20 AR coefficients learned using $\sigma_v^2 = 0.001$, $\sigma_h^2 = 0.001$, see `ARlds.m`. **(e)**: Clustering the results in panel (d) using a Gaussian mixture model with 8 components. The AR components group roughly according to the different song regimes.

Definition 24.2 (Discrete Fourier Transform)**.** For a sequence $x_{0:N-1}$ the DFT $f_{0:N-1}$ is defined as

$$f_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi \mathrm{i}}{N} kn}, \qquad k = 0, \ldots, N-1 \tag{24.2.16}$$

f_k is a (complex) representation as to how much frequency k is present in the sequence $x_{0:N-1}$. The power of component k is defined as the absolute length of the complex f_k.

Definition 24.3 (Spectrogram)**.** Given a timeseries $x_{1:T}$ the spectrogram at time t is a representation of the frequencies present in a window localised around t. For each window one computes the Discrete Fourier Transform, from which we obtain a vector of log power in each frequency. The window is then moved (usually) one step forward and the DFT recomputed. Note that by taking the logarithm, small values in the original signal can translate to visibly appreciable values in the spectrogram.

Example 24.1 (Nightingale)**.** In fig(24.3a) we plot the raw acoustic recording for a 5 second fragment of a nightingale song `freesound.org/samplesViewSingle.php?id=17185`. The spectrogram is also plotted and

gives an indication of which frequencies are present in the signal as a function of time. The nightingale song is very complicated but at least locally can be very repetitive. A crude way to find which segments repeat is to form a cluster analysis of the spectrogram. In fig(24.3c) we show the results of fitting a Gaussian mixture model, section(20.3), with 8 components, from which we see there is some repetition of components locally in time. An alternative representation of the signal is given by the time-varying AR coefficients, section(24.2.3), as plotted in fig(24.3d). A GMM clustering with 8 components fig(24.3e) in this case produces a somewhat clearer depiction of the different phases of the nightingale singing than that afforded by the spectrogram.

24.2.4 Time-varying variance AR models

In the standard AR model, equation (24.2.1), the variance σ^2 is assumed fixed throughout time. For some applications, particular in finance, this is undesirable since the 'volatility' can change dramatically with time. A simple extension of the AR model is to write

$$v_t = \sum_{l=1}^{L} a_l v_{t-l} + \eta_t, \qquad \eta_t \sim \mathcal{N}\left(\eta_t | \mu, \sigma_t^2\right) \tag{24.2.17}$$

$$\bar{v}_t = \sum_{l=1}^{L} a_l v_{t-l} \tag{24.2.18}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{Q} \alpha_i \left(v_{t-i} - \bar{v}_{t-i}\right)^2 \tag{24.2.19}$$

where $\alpha_i \geq 0$. The motivation is that equation (24.2.19) represents a noisy estimate of the variance of the noise, based on a weighted sum of the squared discrepancies between the mean prediction $\bar{v}$ and the actual observation v over the previous q timesteps. This is called an AutoRegressive Conditional Heteroskedasticity (ARCH) model [91].

A further extension is to the generalised ARCH (GARCH) model, for which

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{Q} \alpha_i \left(v_{t-i} - \bar{v}_{t-i}\right)^2 + \sum_{i=1}^{P} \beta_i \sigma_{t-i}^2 \tag{24.2.20}$$

where $\beta_i \geq 0$. One can interpret these as deterministic latent variable models, see section(26.4), albeit with the extension to a higher-order Markov process. In this sense, learning of the parameters by Maximum Likelihood is straightforward since derivatives of the log likelihood can be computed by deterministic propagation, as described in section(26.4). Indeed, based on this insight, a whole range of possible non-linear 'volatility' models come into view.

24.3 Latent Linear Dynamical Systems

The Latent LDS defines a stochastic linear dynamical system in a latent (or 'hidden') space on a sequence of vectors $\mathbf{h}_{1:T}$. Each observation $\mathbf{v}_t$ is as linear function of the latent vector $\mathbf{h}_t$. This model is also called a *linear Gaussian state space model*[2]. The model can also be considered a form of LDS on the joint variables $x_t = (v_t, h_t)$, with parts of the vector x_t missing. For this reason we will also refer to this model as a Linear Dynamical System (without the 'latent' prefix).

Definition 24.4 (Latent linear dynamical system)**.**

$$\begin{array}{lll} \mathbf{h}_t = \mathbf{A}_t \mathbf{h}_{t-1} + \boldsymbol{\eta}_t^h & \boldsymbol{\eta}_t^h \sim \mathcal{N}\left(\boldsymbol{\eta}_t^h | \bar{\mathbf{h}}_t, \boldsymbol{\Sigma}_t^H\right) & \text{transition model} \\ \mathbf{v}_t = \mathbf{B}_t \mathbf{h}_t + \boldsymbol{\eta}_t^v & \boldsymbol{\eta}_t^v \sim \mathcal{N}\left(\boldsymbol{\eta}_t^v | \bar{\mathbf{v}}_t, \boldsymbol{\Sigma}_t^V\right) & \text{emission model} \end{array} \tag{24.3.1}$$

[2] These models are also often called Kalman Filters. We avoid this terminology here since the word 'filter' refers to a specific kind of inference and runs the risk of confusing a filtering algorithm with the model itself.

where $\boldsymbol{\eta}_t^h$ and $\boldsymbol{\eta}_t^v$ are noise vectors. $\mathbf{A}_t$ is called the *transition matrix* and $\mathbf{B}_t$ the *emission matrix*. The terms $\bar{\mathbf{h}}_t$ and $\bar{\mathbf{v}}_t$ are the hidden and output bias respectively. The transition and emission models define a first order Markov model

$$p(\mathbf{h}_{1:T}, \mathbf{v}_{1:T}) = p(\mathbf{h}_1)p(\mathbf{v}_1|\mathbf{h}_1)\prod_{t=2}^{T} p(\mathbf{h}_t|\mathbf{h}_{t-1})p(\mathbf{v}_t|\mathbf{h}_t) \tag{24.3.2}$$

with the transitions and emissions given by Gaussian distributions

$$p(\mathbf{h}_t|\mathbf{h}_{t-1}) = \mathcal{N}\left(\mathbf{h}_t\middle|\mathbf{A}_t\mathbf{h}_{t-1} + \bar{\mathbf{h}}_t, \boldsymbol{\Sigma}_t^H\right), \qquad p(\mathbf{h}_1) = \mathcal{N}\left(\mathbf{h}_1\middle|\boldsymbol{\mu}_\pi, \boldsymbol{\Sigma}_\pi\right) \tag{24.3.3}$$

$$p(\mathbf{v}_t|\mathbf{h}_t) = \mathcal{N}\left(\mathbf{v}_t\middle|\mathbf{B}_t\mathbf{h}_t + \bar{\mathbf{v}}_t, \boldsymbol{\Sigma}_t^V\right) \tag{24.3.4}$$

This (latent) LDS is represented as a belief network in fig(24.5) with the extension to higher orders being intuitive. One may also include an external input $\mathbf{o}_t$ at each time, which will add $\mathbf{C}\mathbf{o}_t$ to the mean of the hidden variable and $\mathbf{D}\mathbf{o}_t$ to the mean of the observation.

Explicit expressions for the transition and emission distributions are given below for the time-invariant case with $\bar{\mathbf{v}}_t = \mathbf{0}, \bar{\mathbf{h}}_t = \mathbf{0}$. Each hidden variable is a multidimensional Gaussian distributed vector $\mathbf{h}_t$, with

$$p(\mathbf{h}_t|\mathbf{h}_{t-1}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_H|}}\exp\left(-\frac{1}{2}\left(\mathbf{h}_t - \mathbf{A}\mathbf{h}_{t-1}\right)^\mathsf{T}\boldsymbol{\Sigma}_H^{-1}\left(\mathbf{h}_t - \mathbf{A}\mathbf{h}_{t-1}\right)\right) \tag{24.3.5}$$

which states that $\mathbf{h}_{t+1}$ has a mean equal to $\mathbf{A}\mathbf{h}_t$ with Gaussian fluctuations described by the covariance matrix $\boldsymbol{\Sigma}_H$. Similarly,

$$p(\mathbf{v}_t|\mathbf{h}_t) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_V|}}\exp\left(-\frac{1}{2}\left(\mathbf{v}_t - \mathbf{B}\mathbf{h}_t\right)^\mathsf{T}\boldsymbol{\Sigma}_V^{-1}\left(\mathbf{v}_t - \mathbf{B}\mathbf{h}_t\right)\right) \tag{24.3.6}$$

describes an output $\mathbf{v}_t$ with mean $\mathbf{B}\mathbf{h}_t$ and covariance $\boldsymbol{\Sigma}_V$.

Example 24.2. Consider a dynamical system defined on two dimensional vectors $\mathbf{h}_t$:

$$\mathbf{h}_{t+1} = \mathbf{R}_\theta\mathbf{h}_t, \qquad \text{with} \quad \mathbf{R}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{24.3.7}$$

$\mathbf{R}_\theta$ rotates the vector $\mathbf{h}_t$ through angle θ in one timestep. Under this LDS $\mathbf{h}$ will trace out points on a circle through time. By taking a scalar projection of $\mathbf{h}_t$, for example,

$$v_t = [\mathbf{h}_t]_1 = [1 \;\; 0]^\mathsf{T}\mathbf{h}_t, \tag{24.3.8}$$

the elements v_t, $t = 1, \ldots, T$ describe a sinusoid through time, see fig(24.6). By using a block diagonal $\mathbf{R} = \text{blkdiag}\left(\mathbf{R}_{\theta_1}, \ldots, \mathbf{R}_{\theta_m}\right)$ and taking a scalar projection of the extended $m \times 2$ dimensional $\mathbf{h}$ vector, one can construct a representation of a signal in terms of m sinusoidal components.

24.4 Inference

Given an observation sequence $\mathbf{v}_{1:T}$ we wish to consider filtering and smoothing, as we did for the HMM, section(23.2.1). For the HMM, in deriving the various message passing recursions, we used only the independence structure encoded by the belief network. Since the LDS has the same independence structure as the HMM, we can use the same independence assumptions in deriving the updates for the LDS. However, in implementing them we need to deal with the issue that we now have continuous hidden variables, rather than discrete states. The fact that the distributions are Gaussian means that we can deal with continuous

messages exactly. In translating the HMM message passing equations, we first replace summation with integration. For example, the filtering recursion (23.2.7) becomes

$$p(\mathbf{h}_t|\mathbf{v}_{1:t}) \propto \int_{\mathbf{h}_{t-1}} p(\mathbf{v}_t|\mathbf{h}_t)p(\mathbf{h}_t|\mathbf{h}_{t-1})p(\mathbf{h}_{t-1}|\mathbf{v}_{1:t-1}), \qquad t > 1 \tag{24.4.1}$$

Since the product of two Gaussians is another Gaussian, and the integral of a Gaussian is another Gaussian, the resulting $p(\mathbf{h}_t|\mathbf{v}_{1:t})$ is also Gaussian. This closure property of Gaussians means that we may represent $p(\mathbf{h}_{t-1}|\mathbf{v}_{1:t-1}) = \mathcal{N}(\mathbf{h}_{t-1}|\mathbf{f}_{t-1}, \mathbf{F}_{t-1})$ with mean $\mathbf{f}_{t-1}$ and covariance $\mathbf{F}_{t-1}$. The effect of equation (24.4.1) is equivalent to updating the mean $\mathbf{f}_{t-1}$ and covariance $\mathbf{F}_{t-1}$ into a mean $\mathbf{f}_t$ and covariance $\mathbf{F}_t$ for $p(\mathbf{h}_t|\mathbf{v}_{1:t})$. Our task below is to find explicit algebraic formulae for these updates.

Numerical stability

Translating the message passing inference techniques we developed for the HMM into the LDS is largely straightforward. Indeed, one could simply run a standard sum-product algorithm (albeit for continuous variables), see `demoSumprodGaussCanonLDS.m`. In long timeseries numerical instabilities can build up and may result in grossly inaccurate results, depending on the transition and emission distribution parameters and the method of implementing the message updates. For this reason specialised routines have been developed that are reasonably numerically stable under certain parameter regimes[288]. For the HMM in section(23.2.1), we discussed two alternative methods for smoothing, the parallel β approach, and the sequential γ approach. The β recursion is suitable when the emission and transition covariance entries are small, and the γ recursion usually preferable in the more standard case of small covariance values.

Analytical shortcuts

In deriving the inference recursions we need to frequently multiply and integrate Gaussians. Whilst in principle straightforward, this can be algebraically tedious and, wherever possible, it is useful to appeal to known shortcuts. For example, one can exploit the general result that the linear transform of a Gaussian random variable is another Gaussian random variable. Similarly it is convenient to make use of the conditioning formulae, as well as the dynamics reversal intuition. These results are stated in section(8.4), and below we derive the most useful for our purposes here.

Consider a linear transformation of a Gaussian random variable:

$$\mathbf{y} = \mathbf{M}\mathbf{x} + \boldsymbol{\eta}, \qquad \boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\eta}|\boldsymbol{\mu}, \boldsymbol{\Sigma}), \qquad \mathbf{x} \sim \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) \tag{24.4.2}$$

where $\mathbf{x}$ and $\boldsymbol{\eta}$ are assumed to be generated from independent processes. To find the distribution $p(\mathbf{y})$, one approach would be to write this formally as

$$p(\mathbf{y}) = \int \mathcal{N}(\mathbf{y}|\mathbf{M}\mathbf{x} + \boldsymbol{\mu}, \boldsymbol{\Sigma})\,\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)\,d\mathbf{x} \tag{24.4.3}$$

and carry out the integral (by completing the square). However, since a Gaussian variable under linear transformation is another Gaussian, we can take a shortcut and just find the mean and covariance of the transformed variable. Its mean is given by

$$\langle \mathbf{y} \rangle = \mathbf{M}\langle \mathbf{x} \rangle + \langle \boldsymbol{\eta} \rangle = \mathbf{M}\boldsymbol{\mu}_x + \boldsymbol{\mu} \tag{24.4.4}$$

To find the covariance, consider the displacement of a variable $\mathbf{h}$ from its mean, which we write as

$$\Delta\mathbf{h} \equiv \mathbf{h} - \langle \mathbf{h} \rangle \tag{24.4.5}$$

The covariance is, by definition, $\left\langle \Delta\mathbf{h}\Delta\mathbf{h}^\mathsf{T} \right\rangle$. For $\mathbf{y}$, the displacement is

$$\Delta\mathbf{y} = \mathbf{M}\Delta\mathbf{x} + \Delta\boldsymbol{\eta}, \tag{24.4.6}$$

So that the covariance is

$$\begin{aligned}\left\langle \Delta\mathbf{y}\Delta\mathbf{y}^\mathsf{T} \right\rangle &= \left\langle (\mathbf{M}\Delta\mathbf{x} + \Delta\boldsymbol{\eta})(\mathbf{M}\Delta\mathbf{x} + \Delta\boldsymbol{\eta})^\mathsf{T} \right\rangle \\ &= \mathbf{M}\left\langle \Delta\mathbf{x}\Delta\mathbf{x}^\mathsf{T} \right\rangle \mathbf{M}^\mathsf{T} + \mathbf{M}\left\langle \Delta\mathbf{x}\Delta\boldsymbol{\eta}^\mathsf{T} \right\rangle + \left\langle \Delta\boldsymbol{\eta}\Delta\mathbf{x}^\mathsf{T} \right\rangle \mathbf{M}^\mathsf{T} + \left\langle \Delta\boldsymbol{\eta}\Delta\boldsymbol{\eta}^\mathsf{T} \right\rangle\end{aligned}$$

Since the noises $\boldsymbol{\eta}$ and $\mathbf{x}$ are assumed independent, $\left\langle \Delta\boldsymbol{\eta}\Delta\mathbf{x}^\mathsf{T} \right\rangle = \mathbf{0}$ we have

$$\boldsymbol{\Sigma}_y = \mathbf{M}\boldsymbol{\Sigma}_x\mathbf{M}^\mathsf{T} + \boldsymbol{\Sigma} \tag{24.4.7}$$

Algorithm 24.1 LDS Forward Pass. Compute the filtered posteriors $p(\mathbf{h}_t|\mathbf{v}_{1:t}) \equiv \mathcal{N}(\mathbf{f}_t, \mathbf{F}_t)$ for a LDS with parameters $\theta_t = \left\{\mathbf{A}, \mathbf{B}, \boldsymbol{\Sigma}^h, \boldsymbol{\Sigma}^v, \bar{\mathbf{h}}, \bar{\mathbf{v}}\right\}_t$. The log-likelihood $L = \log p(\mathbf{v}_{1:T})$ is also returned.

$\{\mathbf{f}_1, \mathbf{F}_1, p_1\} = \textsc{ldsforward}(\mathbf{0}, \mathbf{0}, \mathbf{v}_1; \theta_t)$
$F_0 \leftarrow 0,\ f_0 \leftarrow 0,$
$L \leftarrow \log p_1$
for $t \leftarrow 2, T$ **do**
  $\{\mathbf{f}_t, \mathbf{F}_t, p_t\} = \textsc{ldsforward}(\mathbf{f}_{t-1}, \mathbf{F}_{t-1}, \mathbf{v}_t; \theta)$
  $L \leftarrow L + \log p_t$
end for
function $\textsc{ldsforward}(\mathbf{f}, \mathbf{F}, \mathbf{v}; \theta)$
  $\boldsymbol{\mu}_h \leftarrow \mathbf{A}\mathbf{f} + \bar{\mathbf{h}}, \quad \boldsymbol{\mu}_v \leftarrow \mathbf{B}\boldsymbol{\mu}_h + \bar{\mathbf{v}}$ ▷ Mean of $p(\mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{1:t-1})$
  $\boldsymbol{\Sigma}_{hh} \leftarrow \mathbf{A}\mathbf{F}\mathbf{A}^\mathsf{T} + \boldsymbol{\Sigma}^h, \quad \boldsymbol{\Sigma}_{vv} \leftarrow \mathbf{B}\boldsymbol{\Sigma}_{hh}\mathbf{B}^\mathsf{T} + \boldsymbol{\Sigma}^v, \quad \boldsymbol{\Sigma}_{vh} \leftarrow \mathbf{B}\boldsymbol{\Sigma}_{hh}$ ▷ Covariance of $p(\mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{1:t-1})$
  $\mathbf{f}' \leftarrow \boldsymbol{\mu}_h + \boldsymbol{\Sigma}_{vh}^\mathsf{T}\boldsymbol{\Sigma}_{vv}^{-1}(\mathbf{v} - \boldsymbol{\mu}_v), \quad \mathbf{F}' \leftarrow \boldsymbol{\Sigma}_{hh} - \boldsymbol{\Sigma}_{vh}^\mathsf{T}\boldsymbol{\Sigma}_{vv}^{-1}\boldsymbol{\Sigma}_{vh}$ ▷ Find $p(\mathbf{h}_t|\mathbf{v}_{1:t})$ by conditioning:
  $p' \leftarrow \exp\left(-\frac{1}{2}(\mathbf{v} - \boldsymbol{\mu}_v)^\mathsf{T}\boldsymbol{\Sigma}_{vv}^{-1}(\mathbf{v} - \boldsymbol{\mu}_v)\right) / \sqrt{\det(2\pi\boldsymbol{\Sigma}_{vv})}$ ▷ Compute $p(\mathbf{v}_t|\mathbf{v}_{1:t-1})$
  return $\mathbf{f}', \mathbf{F}', p'$
end function

24.4.1 Filtering

We represent the filtered distribution as a Gaussian with mean $\mathbf{f}_t$ and covariance $\mathbf{F}_t$,

$$p(\mathbf{h}_t|\mathbf{v}_{1:t}) \sim \mathcal{N}(\mathbf{h}_t|\mathbf{f}_t, \mathbf{F}_t) \tag{24.4.8}$$

This is called the *moment representation*. Our task is then to find a recursion for $\mathbf{f}_t, \mathbf{F}_t$ in terms of $\mathbf{f}_{t-1}$, $\mathbf{F}_{t-1}$. A convenient approach is to first find the joint distribution $p(\mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{1:t-1})$ and then condition on $\mathbf{v}_t$ to find the distribution $p(\mathbf{h}_t|\mathbf{v}_{1:t})$. The term $p(\mathbf{h}_t, \mathbf{v}_t|\mathbf{v}_{1:t-1})$ is a Gaussian whose statistics can be found from the relations

$$\mathbf{v}_t = \mathbf{B}\mathbf{h}_t + \boldsymbol{\eta}_t^v, \qquad \mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1} + \boldsymbol{\eta}_t^h \tag{24.4.9}$$

Using the above, and assuming time-invariance and zero biases, we readily find

$$\left\langle \Delta\mathbf{h}_t \Delta\mathbf{h}_t^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle = \mathbf{A}\left\langle \Delta\mathbf{h}_{t-1}\Delta\mathbf{h}_{t-1}^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle \mathbf{A}^\mathsf{T} + \boldsymbol{\Sigma}_H = \mathbf{A}\mathbf{F}_{t-1}\mathbf{A}^\mathsf{T} + \boldsymbol{\Sigma}_H \tag{24.4.10}$$

$$\left\langle \Delta\mathbf{v}_t \Delta\mathbf{h}_t^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle = \mathbf{B}\left\langle \Delta\mathbf{h}_t\Delta\mathbf{h}_t^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle = \mathbf{B}\left(\mathbf{A}\mathbf{F}_{t-1}\mathbf{A}^\mathsf{T} + \boldsymbol{\Sigma}_H\right) \tag{24.4.11}$$

$$\left\langle \Delta\mathbf{v}_t \Delta\mathbf{v}_t^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle = \mathbf{B}\left\langle \Delta\mathbf{h}_t\Delta\mathbf{h}_t^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle \mathbf{B}^\mathsf{T} + \boldsymbol{\Sigma}_V = \mathbf{B}\left(\mathbf{A}\mathbf{F}_{t-1}\mathbf{A}^\mathsf{T} + \boldsymbol{\Sigma}_H\right)\mathbf{B}^\mathsf{T} + \boldsymbol{\Sigma}_V \tag{24.4.12}$$

$$\langle \mathbf{v}_t|\mathbf{v}_{1:t-1}\rangle = \mathbf{B}\mathbf{A}\langle \mathbf{h}_{t-1}|\mathbf{v}_{1:t-1}\rangle, \qquad \langle \mathbf{h}_t|\mathbf{v}_{1:t-1}\rangle = \mathbf{A}\langle \mathbf{h}_{t-1}|\mathbf{v}_{1:t-1}\rangle \tag{24.4.13}$$

In the above, using our moment representation of the forward messages

$$\langle \mathbf{h}_{t-1}|\mathbf{v}_{1:t-1}\rangle \equiv \mathbf{f}_{t-1}, \qquad \left\langle \Delta\mathbf{h}_{t-1}\Delta\mathbf{h}_{t-1}^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle \equiv \mathbf{F}_{t-1} \tag{24.4.14}$$

Then, using conditioning[3] $p(\mathbf{h}_t|\mathbf{v}_t, \mathbf{v}_{1:t-1})$ will have mean

$$\mathbf{f}_t \equiv \langle \mathbf{h}_t|\mathbf{v}_{1:t-1}\rangle + \left\langle \Delta\mathbf{h}_t\Delta\mathbf{v}_t^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle \left\langle \Delta\mathbf{v}_t\Delta\mathbf{v}_t^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle^{-1}(\mathbf{v}_t - \langle \mathbf{v}_t|\mathbf{v}_{1:t-1}\rangle) \tag{24.4.15}$$

and covariance

$$\mathbf{F}_t \equiv \left\langle \Delta\mathbf{h}_t\Delta\mathbf{h}_t^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle - \left\langle \Delta\mathbf{h}_t\Delta\mathbf{v}_t^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle \left\langle \Delta\mathbf{v}_t\Delta\mathbf{v}_t^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle^{-1}\left\langle \Delta\mathbf{v}_t\Delta\mathbf{h}_t^\mathsf{T}|\mathbf{v}_{1:t-1}\right\rangle \tag{24.4.16}$$

[3] $p(\mathbf{x}|\mathbf{y})$ is a Gaussian with mean $\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(\mathbf{y} - \boldsymbol{\mu}_y)$ and covariance $\boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}$.

Writing out the above explicitly we have for the mean:

$$\mathbf{f}_t = \mathbf{A}\mathbf{f}_{t-1} + \mathbf{P}\mathbf{B}^\mathsf{T}\left(\mathbf{B}\mathbf{P}\mathbf{B}^\mathsf{T} + \boldsymbol{\Sigma}_V\right)^{-1}(\mathbf{v}_t - \mathbf{B}\mathbf{A}\mathbf{f}_{t-1}) \tag{24.4.17}$$

and covariance

$$\mathbf{F}_t = \mathbf{P} + \boldsymbol{\Sigma}_H - \mathbf{P}\mathbf{B}^\mathsf{T}\left(\mathbf{B}\mathbf{P}\mathbf{B}^\mathsf{T} + \boldsymbol{\Sigma}_V\right)^{-1}\mathbf{B}\mathbf{P} \tag{24.4.18}$$

where

$$\mathbf{P} \equiv \mathbf{A}\mathbf{F}_{t-1}\mathbf{A}^\mathsf{T} + \boldsymbol{\Sigma}_H \tag{24.4.19}$$

The filtering procedure is presented in algorithm(24.1) with a single update in `LDSforwardUpdate.m`.

One can write the covariance update as

$$\mathbf{F}_t = (\mathbf{I} - \mathbf{K}\mathbf{B})\,\mathbf{P} \tag{24.4.20}$$

where we define the *Kalman gain* matrix

$$\mathbf{K} = \mathbf{P}\mathbf{B}^\mathsf{T}\left(\boldsymbol{\Sigma}_V + \mathbf{B}\mathbf{P}\mathbf{B}^\mathsf{T}\right)^{-1} \tag{24.4.21}$$

The iteration is expected to be numerically stable when the noise covariances are small.

Symmetrising the updates

A potential numerical issue with the covariance update (24.4.20) is that it is the difference of two positive definite matrices. If there are numerical errors, the $\mathbf{F}_t$ may not be positive definite, nor symmetric. Using the Woodbury identity, definition(29.11), equation (24.4.18) can be written more compactly as

$$\mathbf{F}_t = \left(\mathbf{P}^{-1} + \mathbf{B}^\mathsf{T}\boldsymbol{\Sigma}_V^{-1}\mathbf{B}\right)^{-1} \tag{24.4.22}$$

Whilst this is positive semidefinite, this is numerically expensive since it involves two matrix inversions. An alternative is to use the definition of $\mathbf{K}$, from which we can write

$$\mathbf{K}\boldsymbol{\Sigma}_V\mathbf{K}^\mathsf{T} = (\mathbf{I} - \mathbf{K}\mathbf{B})\,\mathbf{P}\mathbf{B}^\mathsf{T}\mathbf{K}^\mathsf{T} \tag{24.4.23}$$

Hence we arrive at *Joseph's symmetrized update*[107]

$$(\mathbf{I} - \mathbf{K}\mathbf{B})\,\mathbf{P}\,(\mathbf{I} - \mathbf{K}\mathbf{B})^\mathsf{T} + \mathbf{K}\boldsymbol{\Sigma}_V\mathbf{K}^\mathsf{T} = (\mathbf{I} - \mathbf{K}\mathbf{B})\left(\mathbf{P}\,(\mathbf{I} - \mathbf{K}\mathbf{B})^\mathsf{T} + \mathbf{P}\mathbf{B}^\mathsf{T}\mathbf{K}^\mathsf{T}\right) = (\mathbf{I} - \mathbf{K}\mathbf{B})\,\mathbf{P} \tag{24.4.24}$$

The left hand side is the addition of two positive definite matrices so that the resulting update for the covariance is more numerically stable. A similar method can be used in the backward pass below. An alternative is to avoid using covariance matrices directly and use their square root as the parameter, deriving updates for these instead[228, 38].

24.4.2 Smoothing : Rauch-Tung-Striebel correction method

The smoothed posterior $p(\mathbf{h}_t|\mathbf{v}_{1:T})$ is necessarily Gaussian since it is the conditional marginal of a larger Gaussian. By representing the posterior as a Gaussian with mean $\mathbf{g}_t$ and covariance $\mathbf{G}_t$,

$$p(\mathbf{h}_t|\mathbf{v}_{1:T}) \sim \mathcal{N}(\mathbf{h}_t|\mathbf{g}_t, \mathbf{G}_t) \tag{24.4.25}$$

we can form a recursion for $\mathbf{g}_t$ and $\mathbf{G}_t$ as follows:

$$p(\mathbf{h}_t|\mathbf{v}_{1:T}) = \int_{\mathbf{h}_{t+1}} p(\mathbf{h}_t, \mathbf{h}_{t+1}|\mathbf{v}_{1:T}) \tag{24.4.26}$$

$$= \int_{\mathbf{h}_{t+1}} p(\mathbf{h}_t|\mathbf{v}_{1:T}, \mathbf{h}_{t+1})p(\mathbf{h}_{t+1}|\mathbf{v}_{1:T}) = \int_{\mathbf{h}_{t+1}} p(\mathbf{h}_t|\mathbf{v}_{1:t}, \mathbf{h}_{t+1})p(\mathbf{h}_{t+1}|\mathbf{v}_{1:T}) \tag{24.4.27}$$

The term $p(\mathbf{h}_t|\mathbf{v}_{1:t}, \mathbf{h}_{t+1})$ can be found by conditioning the joint distribution

$$p(\mathbf{h}_t, \mathbf{h}_{t+1}|\mathbf{v}_{1:t}) = p(\mathbf{h}_{t+1}|\mathbf{h}_t, \cancel{\mathbf{v}_{1:t}})p(\mathbf{h}_t|\mathbf{v}_{1:t}) \tag{24.4.28}$$

which is obtained in the usual manner by finding its mean and covariance: The term $p(\mathbf{h}_t|\mathbf{v}_{1:t})$ is a known Gaussian from filtering with mean $\mathbf{f}_t$ and covariance $\mathbf{F}_t$. Hence the joint distribution $p(\mathbf{h}_t, \mathbf{h}_{t+1}|\mathbf{v}_{1:t})$ has means

$$\langle \mathbf{h}_t|\mathbf{v}_{1:t}\rangle = \mathbf{f}_t, \qquad \langle \mathbf{h}_{t+1}|\mathbf{v}_{1:t}\rangle = \mathbf{A}\mathbf{f}_t \tag{24.4.29}$$

and covariance elements

$$\left\langle \Delta\mathbf{h}_t \Delta\mathbf{h}_t^\mathsf{T}|\mathbf{v}_{1:t}\right\rangle = \mathbf{F}_t, \quad \left\langle \Delta\mathbf{h}_t \Delta\mathbf{h}_{t+1}^\mathsf{T}|\mathbf{v}_{1:t}\right\rangle = \mathbf{F}_t\mathbf{A}^\mathsf{T}, \quad \left\langle \Delta\mathbf{h}_{t+1} \Delta\mathbf{h}_{t+1}^\mathsf{T}|\mathbf{v}_{1:t}\right\rangle = \mathbf{A}\mathbf{F}_t\mathbf{A}^\mathsf{T} + \mathbf{\Sigma}_H \tag{24.4.30}$$

To find $p(\mathbf{h}_t|\mathbf{v}_{1:t}, \mathbf{h}_{t+1})$ we may use the conditioned Gaussian results, result(8.3). It is useful to use the system reversal result, section(8.4.2), which interprets $p(\mathbf{h}_t|\mathbf{v}_{1:t}, \mathbf{h}_{t+1})$ as an equivalent linear system going backwards in time:

$$\mathbf{h}_t = \overleftarrow{\mathbf{A}}_t \mathbf{h}_{t+1} + \overleftarrow{\mathbf{m}}_t + \overleftarrow{\boldsymbol{\eta}}_t \tag{24.4.31}$$

where

$$\overleftarrow{\mathbf{A}}_t \equiv \left\langle \Delta\mathbf{h}_t \Delta\mathbf{h}_{t+1}^\mathsf{T}|\mathbf{v}_{1:t}\right\rangle \left\langle \Delta\mathbf{h}_{t+1} \Delta\mathbf{h}_{t+1}^\mathsf{T}|\mathbf{v}_{1:t}\right\rangle^{-1} \tag{24.4.32}$$

$$\overleftarrow{\mathbf{m}}_t \equiv \langle \mathbf{h}_t|\mathbf{v}_{1:t}\rangle - \left\langle \Delta\mathbf{h}_t \Delta\mathbf{h}_{t+1}^\mathsf{T}|\mathbf{v}_{1:t}\right\rangle \left\langle \Delta\mathbf{h}_{t+1} \Delta\mathbf{h}_{t+1}^\mathsf{T}|\mathbf{v}_{1:t}\right\rangle^{-1} \langle \mathbf{h}_{t+1}|\mathbf{v}_{1:t}\rangle \tag{24.4.33}$$

and $\overleftarrow{\boldsymbol{\eta}}_t \sim \mathcal{N}\left(\overleftarrow{\boldsymbol{\eta}}_t|\mathbf{0}, \overleftarrow{\mathbf{\Sigma}}_t\right)$, with

$$\overleftarrow{\mathbf{\Sigma}}_t \equiv \left\langle \Delta\mathbf{h}_t \Delta\mathbf{h}_t^\mathsf{T}|\mathbf{v}_{1:t}\right\rangle - \left\langle \Delta\mathbf{h}_t \Delta\mathbf{h}_{t+1}^\mathsf{T}|\mathbf{v}_{1:t}\right\rangle \left\langle \Delta\mathbf{h}_{t+1} \Delta\mathbf{h}_{t+1}^\mathsf{T}|\mathbf{v}_{1:t}\right\rangle^{-1} \left\langle \Delta\mathbf{h}_{t+1} \Delta\mathbf{h}_t^\mathsf{T}|\mathbf{v}_{1:t}\right\rangle \tag{24.4.34}$$

Using dynamics reversal, equation (24.4.31) and assuming that $\mathbf{h}_{t+1}$ is Gaussian distributed, it is then straightforward to work out the statistics of $p(\mathbf{h}_t|\mathbf{v}_{1:T})$. The mean is given by

$$\mathbf{g}_t \equiv \langle \mathbf{h}_t|\mathbf{v}_{1:T}\rangle = \overleftarrow{\mathbf{A}}_t \langle \mathbf{h}_{t+1}|\mathbf{v}_{1:T}\rangle + \overleftarrow{\mathbf{m}}_t \equiv \overleftarrow{\mathbf{A}}_t \mathbf{g}_{t+1} + \overleftarrow{\mathbf{m}}_t \tag{24.4.35}$$

and covariance

$$\mathbf{G}_t \equiv \left\langle \Delta\mathbf{h}_t \Delta\mathbf{h}_t^\mathsf{T}|\mathbf{v}_{1:T}\right\rangle = \overleftarrow{\mathbf{A}}_t \left\langle \Delta\mathbf{h}_{t+1} \Delta\mathbf{h}_{t+1}^\mathsf{T}|\mathbf{v}_{1:T}\right\rangle \overleftarrow{\mathbf{A}}_t^\mathsf{T} + \overleftarrow{\mathbf{\Sigma}}_t \equiv \overleftarrow{\mathbf{A}}_t \mathbf{G}_{t+1} \overleftarrow{\mathbf{A}}_t^\mathsf{T} + \overleftarrow{\mathbf{\Sigma}}_t \tag{24.4.36}$$

This procedure is the Rauch-Tung-Striebel Kalman smoother[233]. This is called a 'correction' method since it takes the filtered estimate $p(\mathbf{h}_t|\mathbf{v}_{1:t})$ and 'corrects' it to form a smoothed estimate $p(\mathbf{h}_t|\mathbf{v}_{1:T})$. The procedure is outlined in algorithm(24.2) and is detailed in `LDSbackwardUpdate.m`. See also `LDSsmooth.m`.

Sampling a trajectory from $p(\mathbf{h}_{1:T}|\mathbf{v}_{1:T})$

One may use equation (24.4.31) to draw a sample trajectory $\mathbf{h}_{1:T}$ from the posterior $p(\mathbf{h}_{1:T}|\mathbf{v}_{1:T})$. We start by first drawing a sample $\mathbf{h}_T$ from the Gaussian posterior $p(\mathbf{h}_T|\mathbf{v}_{1:T}) = \mathcal{N}(\mathbf{h}_T|\mathbf{f}_T, \mathbf{F}_T)$. Given this value, we write

$$\mathbf{h}_{T-1} = \overleftarrow{\mathbf{A}}_{T-1} \mathbf{h}_T + \overleftarrow{\mathbf{m}}_{T-1} + \overleftarrow{\boldsymbol{\eta}}_{T-1} \tag{24.4.37}$$

By drawing a sample from the zero mean Gaussian $\mathcal{N}\left(\overleftarrow{\boldsymbol{\eta}}_{T-1}|\mathbf{0}, \overleftarrow{\mathbf{\Sigma}}_{T-1}\right)$, we then have a value for $\mathbf{h}_{T-1}$. We continue in this way, reversing backwards in time, to build up the sample trajectory $\mathbf{h}_T, \mathbf{h}_{T-1}, \mathbf{h}_{T-2}, \ldots, \mathbf{h}_1$. This is equivalent to the forward-filtering-backward-sampling approach described in section(23.2.5).

Algorithm 24.2 LDS Backward Pass. Compute the smoothed posteriors $p(\mathbf{h}_t|\mathbf{v}_{1:T})$. This requires the filtered results from algorithm(24.1).

$\mathbf{G}_T \leftarrow \mathbf{F}_T$, $\mathbf{g}_T \leftarrow \mathbf{f}_T$
for $t \leftarrow T-1, 1$ **do**
  $\{\mathbf{g}_t, \mathbf{G}_t\} = \text{LDSBACKWARD}(\mathbf{g}_{t+1}, \mathbf{G}_{t+1}, \mathbf{f}_t, \mathbf{F}_t; \theta)$
end for
function LDSBACKWARD$(\mathbf{g}, \mathbf{G}, \mathbf{f}, \mathbf{F}; \theta)$
  $\boldsymbol{\mu}_h \leftarrow \mathbf{A}\mathbf{f} + \bar{\mathbf{h}}$, $\quad \boldsymbol{\Sigma}_{h'h'} \leftarrow \mathbf{A}\mathbf{F}\mathbf{A}^\mathsf{T} + \boldsymbol{\Sigma}^h$, $\quad \boldsymbol{\Sigma}_{h'h} \leftarrow \mathbf{A}\mathbf{F}$ ▷ Statistics of $p(\mathbf{h}_t, \mathbf{h}_{t+1}|\mathbf{v}_{1:t})$
  $\overleftarrow{\boldsymbol{\Sigma}} \leftarrow \mathbf{F} - \boldsymbol{\Sigma}_{h'h}^\mathsf{T}\boldsymbol{\Sigma}_{h'h'}^{-1}\boldsymbol{\Sigma}_{h'h}$, $\quad \overleftarrow{\mathbf{A}} \leftarrow \boldsymbol{\Sigma}_{h'h}^\mathsf{T}\boldsymbol{\Sigma}_{h'h'}^{-1}$, $\quad \overleftarrow{\mathbf{m}} \leftarrow \mathbf{f} - \overleftarrow{\mathbf{A}}\boldsymbol{\mu}_h$ ▷ Dynamics Reversal $p(\mathbf{h}_t|\mathbf{h}_{t+1}, \mathbf{v}_{1:t})$
  $\mathbf{g}' \leftarrow \overleftarrow{\mathbf{A}}\mathbf{g} + \overleftarrow{\mathbf{m}}$, $\quad \mathbf{G}' \leftarrow \overleftarrow{\mathbf{A}}\mathbf{G}\overleftarrow{\mathbf{A}}^\mathsf{T} + \overleftarrow{\boldsymbol{\Sigma}}$ ▷ Backward propagation
  return $\mathbf{g}', \mathbf{G}'$
end function

The cross moment

An advantage of the dynamics reversal interpretation given above is that the cross moment (which is required for learning) is immediately obtained from

$$\left\langle \Delta\mathbf{h}_t \Delta\mathbf{h}_{t+1}^\mathsf{T} | \mathbf{v}_{1:T} \right\rangle = \overleftarrow{\mathbf{A}}_t \mathbf{G}_{t+1} \Rightarrow \left\langle \mathbf{h}_t \mathbf{h}_{t+1}^\mathsf{T} | \mathbf{v}_{1:T} \right\rangle = \overleftarrow{\mathbf{A}}_t \mathbf{G}_{t+1} + \mathbf{g}_t \mathbf{g}_{t+1}^\mathsf{T} \tag{24.4.38}$$

24.4.3 The likelihood

We can compute the likelihood using the decomposition

$$p(\mathbf{v}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{v}_t|\mathbf{v}_{1:t-1}) \tag{24.4.39}$$

in which each conditional $p(\mathbf{v}_t|\mathbf{v}_{1:t-1})$ is a Gaussian in $\mathbf{v}_t$. It is straightforward to show that the term $p(\mathbf{v}_t|\mathbf{v}_{1:t-1})$ has mean and covariance

$$\begin{array}{llll} \boldsymbol{\mu}_1 \equiv \mathbf{B}\boldsymbol{\mu} & \boldsymbol{\Sigma}_1 \equiv \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^\mathsf{T} + \boldsymbol{\Sigma}_V & & t = 1 \\ \boldsymbol{\mu}_t \equiv \mathbf{B}\mathbf{A}\mathbf{f}_{t-1} & \boldsymbol{\Sigma}_t \equiv \mathbf{B}\left(\mathbf{A}\mathbf{F}_{t-1}\mathbf{A}^\mathsf{T} + \boldsymbol{\Sigma}_H\right)\mathbf{B}^\mathsf{T} + \boldsymbol{\Sigma}_V & & t > 1 \end{array} \tag{24.4.40}$$

The log likelihood is then given by

$$\log p(\mathbf{v}_{1:T}) = -\frac{1}{2}\sum_{t=1}^{T}\left[(\mathbf{v}_t - \boldsymbol{\mu}_t)^\mathsf{T}\boldsymbol{\Sigma}_t^{-1}(\mathbf{v}_t - \boldsymbol{\mu}_t) + \log\det(2\pi\boldsymbol{\Sigma}_t)\right] \tag{24.4.41}$$

24.4.4 Most likely state

Since the mode of a Gaussian is equal to its mean, there is no difference between the most probable joint posterior state

$$\underset{\mathbf{h}_{1:T}}{\text{argmax}}\; p(\mathbf{h}_{1:T}|\mathbf{v}_{1:T}) \tag{24.4.42}$$

and the set of most probable marginal states

$$h_t = \underset{\mathbf{h}_t}{\text{argmax}}\; p(\mathbf{h}_t|\mathbf{v}_{1:T}), \qquad t = 1, \ldots, T \tag{24.4.43}$$

Hence the most likely hidden state sequence is equivalent to the smoothed mean sequence.

24.4.5 Time independence and Riccati equations

Both the filtered $\mathbf{F}_t$ and smoothed $\mathbf{G}_t$ covariance recursions are independent of the observations $\mathbf{v}_{1:T}$, depending only on the parameters of the model. This is a general characteristic of linear Gaussian systems. Typically the covariance recursions converge quickly to values that are reasonably constant throughout

the dynamics, with only appreciable differences at the boundaries $t = 1$ and $t = T$. In practice one often drops the time-dependence of the covariances and approximates them with a single time-independent covariance. This approximation dramatically reduces storage requirements. The converged filtered $\mathbf{F}$ satisfies the recursion

$$\mathbf{F} = \mathbf{A}\mathbf{F}\mathbf{A}^\mathsf{T} + \mathbf{\Sigma}_H - \left(\mathbf{A}\mathbf{F}\mathbf{A}^\mathsf{T} + \mathbf{\Sigma}_H\right)\mathbf{B}^\mathsf{T}\left(\mathbf{B}\left(\mathbf{A}\mathbf{F}\mathbf{A}^\mathsf{T} + \mathbf{\Sigma}_H\right)\mathbf{B}^\mathsf{T} + \mathbf{\Sigma}_V\right)^{-1}\mathbf{B}\left(\mathbf{A}\mathbf{F}\mathbf{A}^\mathsf{T} + \mathbf{\Sigma}_H\right) \quad (24.4.44)$$

which can be related to a form of *algebraic Riccati equation*. A technique to solve these equations is to being with setting the covariance to $\mathbf{\Sigma}$. With this, a new $\mathbf{F}$ is found using the right hand side of (24.4.44), and subsequently recursively updated. Alternatively, using the Woodbury identity, the converged covariance satisfies

$$\mathbf{F} = \left(\left(\mathbf{A}\mathbf{F}\mathbf{A}^\mathsf{T} + \mathbf{\Sigma}_H\right)^{-1} + \mathbf{B}^\mathsf{T}\mathbf{\Sigma}_V^{-1}\mathbf{B}\right)^{-1} \quad (24.4.45)$$

although this form is less numerically convenient in forming an iterative solver for $\mathbf{F}$ since it requires two matrix inversions.

Example 24.3 (Newtonian Trajectory Analysis)**.** A toy rocket with unknown mass and initial velocity is launched in the air. In addition, the constant accelerations from the rocket's propulsion system are unknown. It is known is that Newton's laws apply and an instrument can measure the vertical height and horizontal distance of the rocket at each time $x(t), y(t)$ from the origin. Based on noisy measurements of $x(t)$ and $y(t)$, our task is to infer the position of the rocket at each time.

Although this is perhaps most appropriately considered from the using continuous time dynamics, we will translate this into a discrete time approximation. Newton's law states that

$$\frac{d^2}{dt^2}x = \frac{f_x(t)}{m}, \qquad \frac{d^2}{dt^2}y = \frac{f_y(t)}{m} \quad (24.4.46)$$

where m is the mass of the object and $f_x(t), f_y(t)$ are the horizontal and vertical forces respectively. Hence As they stand, these equations are not in a form directly usable in the LDS framework. A naive approach is to reparameterise time to use the variable $\tilde{t}$ such that $t \equiv \tilde{t}\Delta$, where $\tilde{t}$ is integer and Δ is a unit of time. The dynamics is then

$$x((\tilde{t}+1)\Delta) = x(\tilde{t}\Delta) + \Delta x'(\tilde{t}\Delta) \quad (24.4.47)$$

$$y((\tilde{t}+1)\Delta) = y(\tilde{t}\Delta) + \Delta y'(\tilde{t}\Delta) \quad (24.4.48)$$

where $y'(t) \equiv \frac{dy}{dt}$. We can write an update equation for the x' and y' as

$$x'((\tilde{t}+1)\Delta) = x'(\tilde{t}\Delta) + f_x\Delta/m, \qquad y'((\tilde{t}+1)\Delta) = y'(\tilde{t}\Delta) + f_y\Delta/m \quad (24.4.49)$$

These are discrete time difference equations indexed by $\tilde{t}$. The instrument which measures $x(t)$ and $y(t)$ is not completely accurate. For simplicity, we relabel $a_x(t) = f_x(t)/m(t)$, $a_y(t) = f_y(t)/m(t)$ – these accelerations will be assumed to be roughly constant, but unknown :

$$a_x((\tilde{t}+1)\Delta) = a_x(\tilde{t}\Delta) + \eta_x, \quad a_y((\tilde{t}+1)\Delta) = a_y(\tilde{t}\Delta) + \eta_y, \quad (24.4.50)$$

where η_x and η_y are small noise terms. The initial distributions for the accelerations are assumed vague, using a zero mean Gaussian with large variance.

We describe the above model by considering $x(t)$, $x'(t)$, $y(t)$, $y'(t)$, $a_x(t)$, $a_y(t)$ as hidden variables, giving rise to a $H = 6$ dimensional LDS with transition and emission matrices as below:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & \Delta & 0 \\ \Delta & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta \\ 0 & 0 & \Delta & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (24.4.51)$$

We place a large variance on their initial values, and attempt to infer the unknown trajectory. A demonstration is given in fig(24.7). Despite the significant observation noise, the object trajectory can be accurately inferred.

24.5 Learning Linear Dynamical Systems

Whilst in many applications, particularly of underlying known physical processes, the paremeters of the LDS are known, in many machine learning tasks we need to learn the parameters of the LDS based on $\mathbf{v}_{1:T}$. For simplicity we assume that we know the dimensionality H of the LDS.

24.5.1 Identifiability issues

An interesting question is whether we can uniquely identify (learn) the parameters of an LDS. There are always trivial redundancies in the solution obtained by permuting the hidden variables arbitrarily and flipping their signs. To show that there are potentially many more equivalent solutions, consider the following LDS

$$\mathbf{v}_t = \mathbf{B}\mathbf{h}_t + \boldsymbol{\eta}_t^v, \qquad \mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1} + \boldsymbol{\eta}_t^h \tag{24.5.1}$$

We now attempt to transform this original system to a new form which will produce exactly the same outputs $\mathbf{v}_{1:T}$. For an invertible matrix $\mathbf{R}$ we consider

$$\mathbf{R}\mathbf{h}_t = \mathbf{R}\mathbf{A}\mathbf{R}^{-1}\mathbf{R}\mathbf{h}_{t-1} + \mathbf{R}\boldsymbol{\eta}_t^h \tag{24.5.2}$$

which is representable as a new latent dynamics

$$\hat{\mathbf{h}}_t = \hat{\mathbf{A}}\hat{\mathbf{h}}_{t-1} + \hat{\boldsymbol{\eta}}_t^h \tag{24.5.3}$$

where $\hat{\mathbf{A}} \equiv \mathbf{R}\mathbf{A}\mathbf{R}^{-1}$, $\hat{\mathbf{h}}_t \equiv \mathbf{R}\mathbf{h}_t$, $\hat{\boldsymbol{\eta}}_t^h \equiv \mathbf{R}\boldsymbol{\eta}_t^h$. In addition, we can reexpress the outputs to be a function of the transformed $\mathbf{h}$:

$$\mathbf{v}_t = \mathbf{B}\mathbf{R}^{-1}\mathbf{R}\mathbf{h}_t + \boldsymbol{\eta}_t^v = \hat{\mathbf{B}}\hat{\mathbf{h}}_t + \boldsymbol{\eta}_t^v \tag{24.5.4}$$

Hence, provided we place no constraints on $\mathbf{A}$, $\mathbf{B}$ and $\boldsymbol{\Sigma}_H$ there exists an infinite space of equivalent solutions, $\hat{\mathbf{A}} = \mathbf{R}\mathbf{A}\,\mathbf{R}^{-1}$, $\hat{\mathbf{B}} = \mathbf{B}\mathbf{R}^{-1}$, $\hat{\boldsymbol{\Sigma}}_H = \mathbf{R}\boldsymbol{\Sigma}_H\mathbf{R}^{\mathsf{T}}$, all with the same likelihood value. This means that directly interpreting the learned parameters needs to be done with some care. This redundancy can be mitigated by imposing constraints on the parameters.

24.5.2 EM algorithm

For simplicity, we assume we have a single sequence $\mathbf{v}_{1:T}$, to which we wish to fit a LDS using Maximum Likelihood. Since the LDS contains latent variables one approach is to use the EM algorithm. As usual, the M-step of the EM algorithm requires us to maximise the energy

$$\langle \log p(\mathbf{v}_{1:T}, \mathbf{h}_{1:T}) \rangle_{p^{old}(\mathbf{h}_{1:T}|\mathbf{v}_{1:T})} \tag{24.5.5}$$

with respect to the parameters $\mathbf{A}$, $\mathbf{B}$, $\mathbf{a}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}_V$, $\boldsymbol{\Sigma}_H$. Thanks to the form of the LDS the energy decomposes as

$$\langle \log p(\mathbf{h}_1) \rangle_{p^{old}(\mathbf{h}_1|\mathbf{v}_{1:T})} + \sum_{t=2}^{T} \langle \log p(\mathbf{h}_t|\mathbf{h}_{t-1}) \rangle_{p^{old}(\mathbf{h}_t,\mathbf{h}_{t-1}|\mathbf{v}_{1:T})} + \sum_{t=1}^{T} \langle \log p(\mathbf{v}_t|\mathbf{h}_t) \rangle_{p^{old}(\mathbf{h}_t|\mathbf{v}_{1:T})} \tag{24.5.6}$$

It is straightforward to derive that the M-step for the parameters is given by (angled brackets $\langle\cdot\rangle$ denote expectation with respect to the smoothed posterior $p(\mathbf{h}_{1:T}|\mathbf{v}_{1:T})$):

$$\boldsymbol{\mu}^{new} = \langle \mathbf{h}_1 \rangle \tag{24.5.7}$$

$$\boldsymbol{\Sigma}^{new} = \left\langle \mathbf{h}_1 \mathbf{h}_1^\mathsf{T} \right\rangle - \langle \mathbf{h}_1 \rangle \langle \mathbf{h}_1 \rangle^\mathsf{T} \tag{24.5.8}$$

$$\mathbf{A}^{new} = \sum_{t=1}^{T-1} \left\langle \mathbf{h}_{t+1} \mathbf{h}_t^\mathsf{T} \right\rangle \left(\sum_{t=1}^{T-1} \left\langle \mathbf{h}_t \mathbf{h}_t^\mathsf{T} \right\rangle \right)^{-1} \tag{24.5.9}$$

$$\mathbf{B}^{new} = \sum_{t=1}^{T} \mathbf{v}_t \langle \mathbf{h}_t \rangle^\mathsf{T} \left(\sum_{t=1}^{T} \left\langle \mathbf{h}_t \mathbf{h}_t^\mathsf{T} \right\rangle \right)^{-1} \tag{24.5.10}$$

$$\boldsymbol{\Sigma}_V^{new} = \frac{1}{T} \sum_{t=1}^{T} \left(\mathbf{v}_t \mathbf{v}_t^\mathsf{T} - \mathbf{v}_t \langle \mathbf{h}_t \rangle^\mathsf{T} \mathbf{B}^{new\mathsf{T}} - \mathbf{B}^{new} \langle \mathbf{h}_t \rangle \mathbf{v}_t^\mathsf{T} + \mathbf{B}^{new} \left\langle \mathbf{h}_t \mathbf{h}_t^\mathsf{T} \right\rangle \mathbf{B}^{new\mathsf{T}} \right) \tag{24.5.11}$$

$$\boldsymbol{\Sigma}_H^{new} = \frac{1}{T-1} \sum_{t=1}^{T-1} \left(\left\langle \mathbf{h}_{t+1} \mathbf{h}_{t+1}^\mathsf{T} \right\rangle - \mathbf{A}^{new} \left\langle \mathbf{h}_t \mathbf{h}_{t+1}^\mathsf{T} \right\rangle - \left\langle \mathbf{h}_{t+1} \mathbf{h}_t^\mathsf{T} \right\rangle \mathbf{A}^{new\mathsf{T}} + \mathbf{A}^{new} \left\langle \mathbf{h}_t \mathbf{h}_t^\mathsf{T} \right\rangle \mathbf{A}^{new\mathsf{T}} \right) \tag{24.5.12}$$

The above can be simplified to

$$\boldsymbol{\Sigma}_V^{new} = \frac{1}{T} \sum_{t} \left(\mathbf{v}_t \mathbf{v}_t^\mathsf{T} - \mathbf{v}_t \langle \mathbf{h}_t \rangle^\mathsf{T} \mathbf{B}^{new\mathsf{T}} \right) \tag{24.5.13}$$

Similarly,

$$\boldsymbol{\Sigma}_H^{new} = \frac{1}{T-1} \sum_{t=1}^{T-1} \left(\left\langle \mathbf{h}_{t+1} \mathbf{h}_{t+1}^\mathsf{T} \right\rangle - \mathbf{A}^{new} \left\langle \mathbf{h}_t \mathbf{h}_{t+1}^\mathsf{T} \right\rangle \right) \tag{24.5.14}$$

The statistics required therefore include smoothed means, covariances, and cross moments. The extension to learning multiple timeseries is straightforward since the energy is simply summed over the individual sequences. See also exercise(24.6).

The performance of the EM algorithm for the LDS often depends heavily on the initialisation. If we remove the hidden to hidden links, the model is closely related to Factor Analysis (the LDS can be considered a temporal extension of Factor Analysis). One initialisation technique is therefore to learn the $\mathbf{B}$ matrix using Factor Analysis by treating the observations as temporally independent.

Note that whilst the LDS model in not identifiable, as described in section(24.5.1), the M-step for the EM algorithm is unique. This apparent contradiction is resolved when one considers that the EM algorithm is a conditional method, updating depending on the previous parameters and ultimately the initialisation. It is this initialisation that breaks the invariance of the EM algorithm.

24.5.3 Subspace Methods

An alternative to EM and Maximum Likelihood training of an LDS is to use a subspace method[286, 250]. The chief benefit of these techniques is that they avoid the convergence difficulties of EM. To motivate subspace techniques, consider a deterministic LDS

$$\mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1}, \qquad \mathbf{v}_t = \mathbf{B}\mathbf{h}_t \tag{24.5.15}$$

Under this assumption, $\mathbf{v}_t = \mathbf{B}\mathbf{h}_t = \mathbf{B}\mathbf{A}\mathbf{h}_{t-1}$ and, more generally, $\mathbf{v}_t = \mathbf{B}\mathbf{A}^t\mathbf{h}_1$. This means that a low dimensional system underlies all visible information since all points $\mathbf{A}^t\mathbf{h}_1$ lie in a H-dimensional subspace, which is then projected to form the observation. This suggests that some form of subspace identification technique will enable us to learn $\mathbf{A}$ and $\mathbf{B}$.

Given a set of observation vectors $\mathbf{v}_1, \ldots, \mathbf{v}_t$, consider the block *Hankel matrix* formed from stacking the vectors. For an order L matrix, this is a $VL \times T - L + 1$ matrix. For example, for $T = 6$ and $L = 3$, this is

$$\mathbf{M} = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 \\ \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 \\ \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 & \mathbf{v}_6 \end{pmatrix} \tag{24.5.16}$$

If the $\mathbf{v}$ are generated from a (noise free) LDS, we can write

$$\mathbf{M} = \begin{pmatrix} \mathbf{B}\mathbf{h}_1 & \mathbf{B}\mathbf{h}_2 & \mathbf{B}\mathbf{h}_3 & \mathbf{B}\mathbf{h}_4 \\ \mathbf{B}\mathbf{A}\mathbf{h}_1 & \mathbf{B}\mathbf{A}\mathbf{h}_2 & \mathbf{B}\mathbf{A}\mathbf{h}_3 & \mathbf{B}\mathbf{A}\mathbf{h}_4 \\ \mathbf{B}\mathbf{A}^2\mathbf{h}_1 & \mathbf{B}\mathbf{A}^2\mathbf{h}_2 & \mathbf{B}\mathbf{A}^2\mathbf{h}_3 & \mathbf{B}\mathbf{A}^2\mathbf{h}_4 \end{pmatrix} = \begin{pmatrix} \mathbf{B} \\ \mathbf{B}\mathbf{A} \\ \mathbf{B}\mathbf{A}^2 \end{pmatrix} \begin{pmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_4 \end{pmatrix} \tag{24.5.17}$$

We now find the SVD of $\mathbf{M}$,

$$\mathbf{M} = \hat{\mathbf{U}} \underbrace{\hat{\mathbf{S}}\hat{\mathbf{V}}^{\mathsf{T}}}_{\mathbf{W}} \tag{24.5.18}$$

where $\mathbf{W}$ is termed the *extended observability matrix*. The matrix $\hat{\mathbf{S}}$ will contain the singular values up to the dimension of the hidden variables H, with the remaining singular values 0. From equation (24.5.17), this means that the emission matrix $\mathbf{B}$ is contained in $\hat{\mathbf{U}}_{1:V,1:H}$. The estimated hidden variables are then contained in the submatrix $\mathbf{W}_{1:H,1:T-L+1}$,

$$\begin{pmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_4 \end{pmatrix} = \mathbf{W}_{1:H,1:T-L+1} \tag{24.5.19}$$

Based on the relation $\mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1}$ one can then find the best least squares estimate for $\mathbf{A}$ by minimising

$$\sum_{t=2}^{T} (\mathbf{h}_t - \mathbf{A}\mathbf{h}_{t-1})^2 \tag{24.5.20}$$

for which the optimal solution is

$$\mathbf{A} = \begin{pmatrix} \mathbf{h}_2 & \mathbf{h}_3 & \ldots & \mathbf{h}_t \end{pmatrix} \begin{pmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \ldots & \mathbf{h}_{T-1} \end{pmatrix}^{\dagger} \tag{24.5.21}$$

where † denotes the pseudo inverse, see `LDSsubspace.m`. Estimates for the covariance matrices can also be obtained from the residual errors in fitting the block Hankel matrix ($\mathbf{\Sigma}_V$) and extended observability matrix ($\mathbf{\Sigma}_H$). Whilst this derivation formally holds only for the noise free case one can nevertheless apply this in the case of non-zero noise and hope to gain an estimate for $\mathbf{A}$ and $\mathbf{B}$ that is correct in the mean. In addition to forming a solution in its own right, the subspace method forms a potentially useful way to initialise the EM algorithm.

24.5.4 Structured LDSs

Many physical equations are local both in time and space. For example in weather models the atmosphere is partitioned into cells $h_i(t)$ each containing the pressure at that location. The equations describing how the pressure updates only depend on the pressure at the current cell and small number of neighbouring cells at the previous time $t-1$. If we use a linear model and measure some aspects of the cells at each time, then the weather is describable by a LDS with a highly structured sparse transition matrix $\mathbf{A}$. In practice, the weather models are non-linear but local linear approximations are often employed[255]. A similar situation arises in brain imaging in which voxels (local cubes of activity) depend only on their neighbours from the previous timestep[104].

Another application of structured LDSs is in temporal independent component analysis. This is defined as the discovery of a set of independent latent dynamical processes, from which the data is a projected observation. If each independent dynamical process can itself be described by a LDS, this gives rise to a structured LDS with a block diagonal transition matrix $\mathbf{A}$. Such models can be used to extract independent components under prior knowledge of the likely underlying frequencies in each of the temporal compoments[59]. See also exercise(**??**).

24.5.5 Bayesian LDSs

The extension to placing priors on the transition and emission parameters of the LDS leads in general to computational difficulties in computing the likelihood. For example, for a prior on $\mathbf{A}$, the likelihood is $p(\mathbf{v}_{1:T}) = \int_{\mathbf{A}} p(\mathbf{v}_{1:T}|\mathbf{A})p(\mathbf{A})$ which is difficult to evaluate since the dependence of the likelihood on the matrix $\mathbf{A}$ is a complicated function. Approximate treatments of this case are beyond the scope of this book, although we briefly note that sampling methods[55, 100] are popular in this context, in addition to deterministic variational approximations[27, 23, 59].

24.6 Switching Auto-Regressive Models

For a time-series of scalar values $v_{1:T}$ an L^{th} order switching AR model can be written as

$$v_t = \hat{\mathbf{v}}_{t-1}^{\mathsf{T}}\mathbf{a}(s_t) + \eta_t, \qquad \eta_t \sim \mathcal{N}\left(\eta_t|0, \sigma^2(s_t)\right) \tag{24.6.1}$$

where we now have a set of AR coefficients $\theta = \left\{\mathbf{a}(s), \sigma^2(s), s \in \{1, \ldots, S\}\right\}$. The discrete switch variables themselves have a Markov transition $p(s_{1:T}) = \prod_t p(s_t|s_{t-1})$ so that the full model is

$$p(v_{1:T}, s_{1:T}|\theta) = \prod_t p(v_t|v_{t-1}, \ldots, v_{t-L}, s_t, |\theta)p(s_t|s_{t-1}) \tag{24.6.2}$$

24.6.1 Inference

Given an observed sequence $v_{1:T}$ and parameters θ inference is straightforward since this is a form of HMM. To make this more apparent we may write

$$p(v_{1:T}, s_{1:T}) = \prod_t \hat{p}(v_t|s_t)p(s_t|s_{t-1}) \tag{24.6.3}$$

where

$$\hat{p}(v_t|s_t) \equiv p(v_t|v_{t-1}, \ldots, v_{t-L}, s_t) = \mathcal{N}\left(v_t\middle|\hat{\mathbf{v}}_{t-1}^{\mathsf{T}}\mathbf{a}(s_t), \sigma^2(s_t)\right) \tag{24.6.4}$$

Note that the emission distribution $\hat{p}(v_t|s_t)$ is time-dependent. The filtering recursion is then

$$\alpha(s_t) = \sum_{s_{t-1}} \hat{p}(v_t|s_t)p(s_t|s_{t-1})\alpha(s_{t-1}) \tag{24.6.5}$$

Smoothing can be achieved using the standard recursions, modified to use the time-dependent emissions, see `demoSARinference.m`.

24.6.2 Maximum Likelihood Learning using EM

To fit the set of AR coefficients and innovation variances, $\mathbf{a}(s), \sigma^2(s), s = 1, \ldots, S$, using Maximum Likelihood training for a set of data $v_{1:T}$, we may make use of the EM algorithm.

M-step

Up to negligible constants, the energy is given by

$$E = \sum_t \left\langle \log p(v_t|\hat{\mathbf{v}}_{t-1}, \mathbf{a}(s_t)) \right\rangle_{p^{old}(s_t|v_{1:T})} + \sum_t \left\langle \log p(s_t|s_{t-1}) \right\rangle_{p^{old}(s_t, s_{t-1})} \tag{24.6.6}$$

which we need to maximise with respect to the parameters θ. Using the definition of the emission and isolating the dependency on $\mathbf{a}$, we have

$$-2E = \sum_t \left\langle \frac{1}{\sigma^2(s_t)}\left(v_t - \hat{\mathbf{v}}_{t-1}^{\mathsf{T}}\mathbf{a}(s_t)\right)^2 + \log\sigma^2(s_t) \right\rangle_{p^{old}(s_t|v_{1:T})} + \mathsf{const.} \tag{24.6.7}$$

On differentiating with respect to $\mathbf{a}(\mathsf{s})$ and equating to zero, the optimal $\mathbf{a}(\mathsf{s})$ satisfies the linear equation

$$\sum_t p^{old}(s_t = \mathsf{s}|v_{1:T})\frac{v_t\hat{\mathbf{v}}_{t-1}}{\sigma^2(\mathsf{s})} = \left[\sum_t p^{old}(s_t = \mathsf{s}|v_{1:T})\frac{\hat{\mathbf{v}}_{t-1}\hat{\mathbf{v}}_{t-1}^{\mathsf{T}}}{\sigma^2(\mathsf{s})}\right]\mathbf{a}(\mathsf{s}) \tag{24.6.8}$$

which may be solved using Gaussian elimination. Similarly one may show that updates that maximise the energy with respect to σ^2 are

$$\sigma^2(\mathsf{s}) = \frac{1}{\sum_{t'} p^{old}(s'_t = \mathsf{s}|v_{1:T})} \sum_t p^{old}(s_t = \mathsf{s}|v_{1:T}) \left[v_t - \hat{\mathbf{v}}_{t-1}^{\mathsf{T}} \mathbf{a}(s_t)\right]^2 \tag{24.6.9}$$

The update for $p(s_t|s_{t-1})$ follows the standard EM for HMM rule, equation (23.3.5), see `SARlearn.m`. Here we don't include an update for the prior $p(s_1)$ since there is insufficient information at the start of the sequence and assume $p(s_1)$ is flat. With high frequency data it is unlikely that a change in the switch variable is reasonable at each time t. A simple constraint to account for this is to use a modified transition

$$\hat{p}(s_t|s_{t-1}) = \begin{cases} p(s_t|s_{t-1}) & \mod(t, T_{skip}) = 0 \\ \delta(s_t - s_{t-1}) & \text{otherwise} \end{cases} \tag{24.6.10}$$

E-step

The M-step requires the smoothed statistics $p^{old}(s_t = \mathsf{s}|v_{1:T})$ and $p^{old}(s_t = \mathsf{s}, s_{t-1} = \mathsf{s}'|v_{1:T})$ which can be obtained from HMM inference.

Example 24.4 (Learning a switching AR model)**.** In fig(24.9) the training data is generated by an Switching AR model so that we know the ground truth as to which model generated which parts of the data. Based on the training data (assuming the labels s_t are unknown), a Switching AR model is fitted using EM. In this case the problem is straightforward so that a good estimate is obtained of both the sets of AR parameters and which switches were used at which time.

Example 24.5 (Modelling parts of speech)**.** In fig(24.10) a segment of a speech signal is shown described by a Switching AR model. Each of the 10 available AR models is responsible for modelling the dynamics of a basic subunit of speech[92][195]. The model was trained on many example sequences using $S = 10$ states with a left-to-right transition matrix. The interest is to determine when each subunit is most likely to be active. This corresponds to the computation of the most-likely switch path $s_{1:T}$ given the observed signal $p(s_{1:T}|\tilde{v}_{1:T})$.

24.7 Code

In the Linear Dynamical System code below only the simplest form of the recursions is given. No attempt has been made to ensure numerical stability.

`LDSforwardUpdate.m`: LDS forward
`LDSbackwardUpdate.m`: LDS backward
`LDSsmooth.m`: Linear Dynamical System : filtering and smoothing
`LDSforward.m`: Alternative LDS forward algorithm (see SLDS chapter)
`LDSbackward.m`: Alternative LDS backward algorithm (see SLDS chapter)
`demoSumprodGaussCanonLDS.m`: Sum-product algorithm for smoothed inference

`demoLDStracking.m`: Demo of tracking in a Newtonian system

`LDSsubspace.m`: Subspace Learning (Hankel matrix method)
`demoLDSsubspace.m`: Demo of Subspace Learning method

24.7.1 Autoregressive models

Note that in the code the autoregressive vector **a** has as its last entry the first AR coefficient (*i.e.* in reverse order to that presented in the text).
`ARtrain.m`: Learn AR coefficients (Gaussian Elimination)
`demoARtrain.m`: Demo of fitting an AR model to data
`ARlds.m`: Learn AR coefficients using a LDS
`demoARlds.m`: Demo of learning AR coefficients using an LDS
`demoSARinference.m`: Demo for inference in a Switching Autoregressive Model

In `SARlearn.m` a slight fudge is used since we do not deal fully with the case at the start where there is insufficient information to define the AR model. For long timeseries this will have a negligible effect, although it might lead to small decreases in the log likelihood under the EM algorithm.
`SARlearn.m`: Learning of a SAR using EM
`demoSARlearn.m`: Demo of SAR learning
`HMMforwardSAR.m`: Switching Autoregressive HMM forward pass
`HMMbackwardSAR.m`: Switching Autoregressive HMM backward pass

24.8 Summary

- Continuous observations can be modelled using simple autoregressive models.
- Observed linear dynamical systems are the vector versions of autoregressive models.
- Latent continuous dynamical processes can be used to model many physical systems. In order for these to be computationally tractable, one may assume that the transitions and observations are linear updates with Gaussian additive noise.
- The latent linear dynamical system is a powerful timeseries model with widespread application in tracking and signal representation.

24.9 Exercises

Exercise 24.1. *Consider the two-dimension linear model*

$$\mathbf{h}_t = \mathbf{R}_\theta \mathbf{h}_{t-1}, \qquad \mathbf{R}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{24.9.1}$$

$\mathbf{R}_\theta$ *is rotation matrix which rotates the vector* $\mathbf{h}_t$ *through angle* θ *in one timestep.*

1. *By writing*

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \end{pmatrix} \tag{24.9.2}$$

   eliminate y_t *to write an equation for* x_{t+1} *in terms of* x_t *and* x_{t-1}.

2. *Explain why the eigenvalues of a rotation matrix are (in general) imaginary.*

3. *Explain how to model a sinusoid, rotating with angular velocity* ω *using a two-dimensional LDS.*

4. *Explain how to model a sinusoid using an AR model.*

5. *Explain the relationship between the second order differential equation $\ddot{x} = -\lambda x$, which describes a Harmonic Oscillator, and the second order difference equation which approximates this differential equation. Is it possible to find a difference equation which exactly matches the solution of the differential equation at chosen points?*

Exercise 24.2. *Show that for any anti-symmetric matrix* $\mathbf{M}$*,*

$$\mathbf{M} = -\mathbf{M}^T \tag{24.9.3}$$

the matrix exponential (in MATLAB *this is* `expm`*)*

$$\mathbf{A} = e^{\mathbf{M}} \tag{24.9.4}$$

is orthogonal, namely

$$\mathbf{A}^T\mathbf{A} = \mathbf{I} \tag{24.9.5}$$

Explain how to construct random orthogonal matrices with some control over the angles of the complex eigenvalues.

Exercise 24.3. *Run* `demoLDStracking.m` *which tracks a ballistic object using a Linear Dynamical system, see example(24.3). Modify* `demoLDStracking.m` *so that in addition to the x and y positions, the x speed is also observed. Compare and contrast the accuracy of the tracking with and without this extra information.*

Exercise 24.4. `nightsong.mat` *contains a small stereo segment nightingale song sampled at* 44100 *Hertz.*

1. *Plot the original waveform using* `plot(x(:,1))`
2. *Download the program* `myspecgram.m` *from* `labrosa.ee.columbia.edu/matlab/sgram/myspecgram.m` *and plot the spectrogram*

   ```
   y=myspecgram(x(:,1),1024,44100); imagesc(log(abs(y)))
   ```

3. *The routine* `demoGMMem.m` *demonstrates fitting a mixture of Gaussians to data. The mixture assignment probabilities are contained in* `phgn`*. Write a routine to cluster the data* `v=log(abs(y))` *using 8 Gaussian components, and explain how one might segment the series* `x` *into different regions.*
4. *Examine* `demoARlds.m` *which fits autoregressive coefficients using an interpretation as a Linear Dynamical System. Adapt the routine* `demoARlds.m` *to learn the AR coefficients of the data* `x`*. You will almost certainly need to subsample the data* `x` *– for example by taking every 4^{th} datapoint. With the learned AR coefficients (use the smoothed results) fit a Gaussian mixture with 8 components. Compare and contrast your results with those obtained from the Gaussian mixture model fit to the spectrogram.*

Exercise 24.5. *Consider a supervised learning problem in which we make a linear model of the scaler output y_t based on vector input $\mathbf{x}_t$:*

$$y_t = \mathbf{w}_t^T\mathbf{x}_t + \boldsymbol{\eta}_t^y \tag{24.9.6}$$

where $\boldsymbol{\eta}_t^y$ is zero mean Gaussian noise. Training data $\mathcal{D} = \{(\mathbf{x}_t, y_t), t = 1, \ldots, T\}$ is available.

1. *For a time-invariant weight vector $\mathbf{w}_t \equiv \mathbf{w}$, explain how to find the single weight vector $\mathbf{w}$ and the noise variance σ^2 by Maximum Likelihood.*
2. *Extend the above model to include a transition*

   $$\mathbf{w}_t = \mathbf{w}_{t-1} + \boldsymbol{\eta}_t^w \tag{24.9.7}$$

   where $\boldsymbol{\eta}_t^w$ is zero mean Gaussian noise with a given covariance $\boldsymbol{\Sigma}$; $\mathbf{w}_1$ has zero mean. Explain how to cast finding $\langle \mathbf{w}_t|\mathcal{D}\rangle$ as smoothing in a related Linear Dynamical System. Write a routine `W = LinPredAR(X,Y,SigmaW,SigmaY)` *that takes an input data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T]$ where each column contains an input, and vector $Y = [y_1, \ldots, y_T]^T$;* `SigmaW` *is the additive weight noise and* `SigmaY` *is an assumed known time-invariant output noise. The returned* `W` *contains the smoothed mean weights.*

Exercise 24.6. *Consider a latent LDS additionally with an observation mean,* $\mathbf{b}$*,*

$$\mathbf{v}_t = \mathbf{B}\mathbf{h}_t + \mathbf{b} + \boldsymbol{\eta}_t^v \tag{24.9.8}$$

Show that for the EM algorithm, the M-step optimally sets $\mathbf{b}$ *to the mean of the observations.*

Exercise 24.7. *This exercise relates to forming an* α*-*β *style smoothing approach, as described in section(23.2.3), but applied to the LDS. Note that the derivations in section(23.2.3) hold for continuous variables as well simply on replacing summation with integration.*

1. *By virtue of the fact that the smoothed posterior for a LDS,* $\gamma(\mathbf{h}_t) \equiv p(\mathbf{h}_t|\mathbf{v}_{1:T})$ *is Gaussian, and using the relation* $\gamma(\mathbf{h}_t) = \alpha(\mathbf{h}_t)\beta(\mathbf{h}_t)$*, explain why the* β *message for an LDS can be represented in the form*

$$\beta(\mathbf{h}_t) = z_t e^{-\frac{1}{2}\mathbf{h}_t^T\mathbf{Z}_t\mathbf{h}_t + \mathbf{h}_t^T\mathbf{z}_t} \tag{24.9.9}$$

   where $\mathbf{Z}_t$ *is a (not necessarily) full rank matrix.*

2. *Based on the recursion*

$$\beta(\mathbf{h}_{t-1}) = \int_{\mathbf{h}_t} p(\mathbf{v}_t|\mathbf{h}_t)p(\mathbf{h}_t|\mathbf{h}_{t-1})\beta(\mathbf{h}_t) \tag{24.9.10}$$

   derive the recursion (ignoring the prefactor z_t*)*

$$\mathbf{L}_t = \mathbf{Q}_t^{-1} + \mathbf{B}_t^T\mathbf{R}_t^{-1}\mathbf{B}_t + \mathbf{Z}_t \tag{24.9.11}$$

$$\mathbf{Z}_{t-1} = \mathbf{A}_t^T\left(\mathbf{Q}_t^{-1} - \mathbf{Q}_t^{-1}\mathbf{L}_t^{-1}\mathbf{Q}_t^{-1}\right)\mathbf{A}_t \tag{24.9.12}$$

$$\mathbf{z}_{t-1} = \mathbf{A}_t^T\mathbf{Q}_t^{-1}\mathbf{L}_t^{-1}\left(\mathbf{B}_t^T\mathbf{R}_t^{-1} + \mathbf{z}_t\right) \tag{24.9.13}$$

   where $\mathbf{Z}_T = \mathbf{0}$*,* $\mathbf{z}_T = \mathbf{0}$*. The notation* $\mathbf{Q}_t \equiv \boldsymbol{\Sigma}_t^H$ *is the covariance matrix of the transition distribution* $p(\mathbf{h}_t|\mathbf{h}_{t-1})$*, and* $\mathbf{R}_t \equiv \boldsymbol{\Sigma}_t^V$ *is the covariance of the emission* $p(\mathbf{v}_t|\mathbf{h}_t)$*.*

3. *Show that the posterior covariance and mean are given by*

$$\left(\mathbf{F}_t^{-1} + \mathbf{Z}_t\right)^{-1}, \qquad \left(\mathbf{F}_t^{-1} + \mathbf{Z}_t\right)^{-1}\left(\mathbf{F}_t^{-1}\mathbf{f}_t + \mathbf{z}_t\right) \tag{24.9.14}$$

   where $\mathbf{F}_t$ *and* $\mathbf{f}_t$ *are the filtered mean and covariance.*

Note that this parallel smoothing recursion is not appropriate in the case of small covariance since, due to the explicit appearance of the inverse of the covariances, numerical stability issues can arise. However, it is possible to reexpress the recursion without explicit reference to inverse noise covariances, see [16].

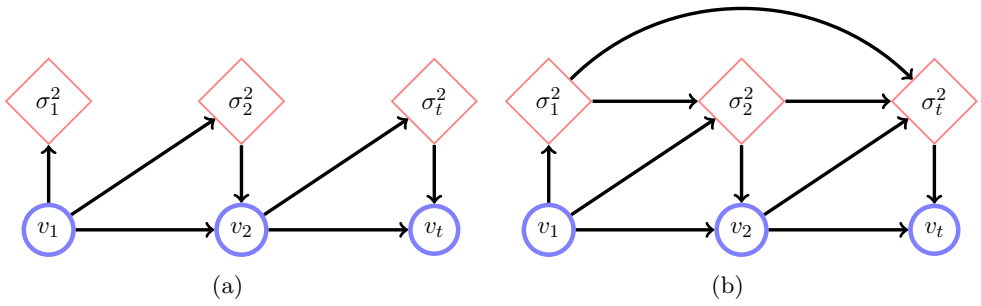

Figure 24.4: **(a)**: A first order $L = 1$, $Q = 1$ ARCH model, in which the observations are dependent on the previous observation, and the variance is dependent on the previous observations in a deterministic manner. **(b)**: An $L = 1$, $Q = 1$, $P = 2$ GARCH model, in which the observations are dependent on the previous observation, and the variance is dependent on the previous observations and previous two variances in a deterministic manner. These are special cases of the general deterministic latent variable models, section(26.4).

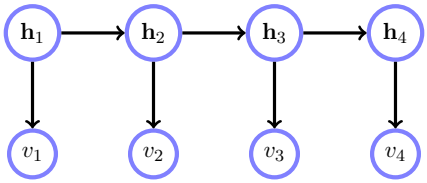

Figure 24.5: A (latent) LDS. Both hidden and visible variables are Gaussian distributed.

Figure 24.6: A single phasor plotted as a damped two dimensional rotation $\mathbf{h}_{t+1} = \gamma \mathbf{R}_\theta \mathbf{h}_t$ with a damping factor $0 < \gamma < 1$. By taking a projection onto the y axis, the phasor generates a damped sinusoid.

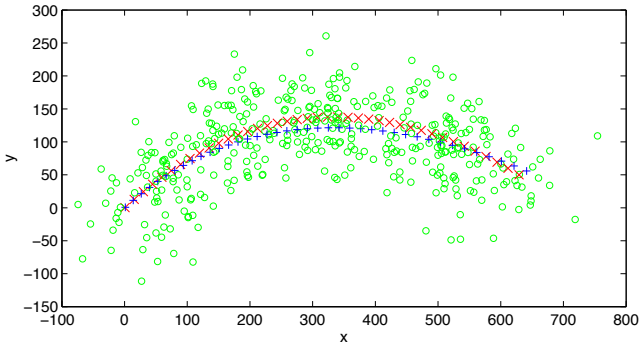

Figure 24.7: Estimate of the trajectory of a Newtonian ballistic object based on noisy observations (small circles). All time labels are known but omitted in the plot. The 'x' points are the true positions of the object, and the crosses '+' are the estimated smoothed mean positions $\langle x_t, y_t | \mathbf{v}_{1:T} \rangle$ of the object plotted every several time steps. See `demoLDStracking.m`

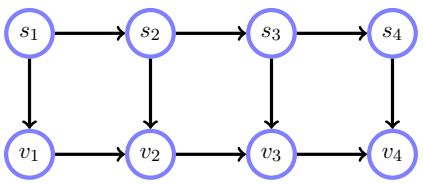

Figure 24.8: A first order switching AR model. In terms of inference, conditioned on $v_{1:T}$, this is a HMM.

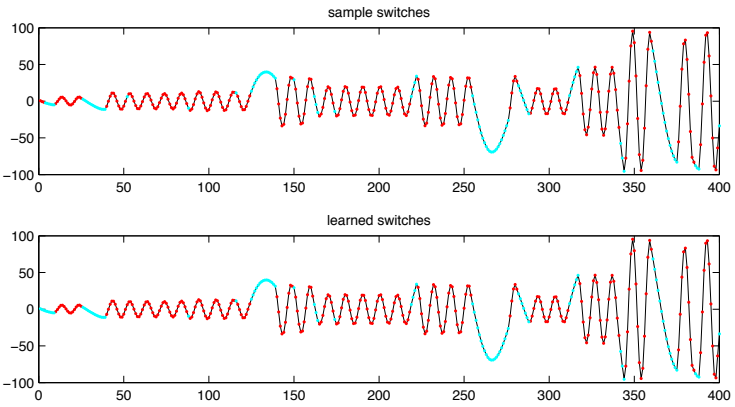

Figure 24.9: Learning a Switching AR model. The upper plot shows the training data. The colour indicates which of the two AR models is active at that time. Whilst this information is plotted here, this is assumed unknown to the learning algorithm, as are the coefficients $\mathbf{a}(s)$. We assume that the order $L = 2$ and number of switches $S = 2$ however is known. In the bottom plot we show the time series again after training in which we colour the points according to the most likely smoothed AR model at each timestep. See `demoSARlearn.m`.

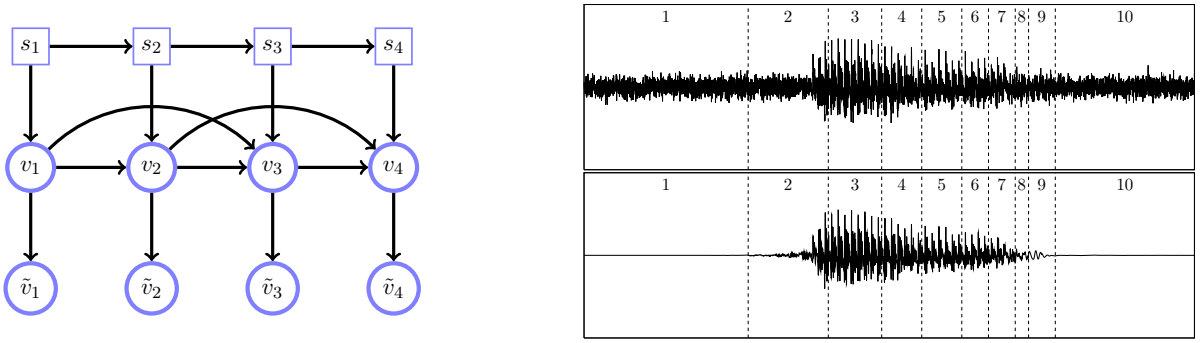

Figure 24.10: **(a)**: A latent switching (second order) AR model. Here the s_t indicates which of a set of 10 available AR models is active at time t. The square nodes emphasise that these are discrete variables. The 'clean' AR signal v_t, which is not observed, is corrupted by additive noise to form the noisy observations $\tilde{v}_t$. In terms of inference, conditioned on $\tilde{v}_{1:T}$, this can be expressed as a Switching LDS, chapter(25). **(b)**: Signal reconstruction using the latent switching AR model in (a). Top: noisy signal $\tilde{v}_{1:T}$; bottom: reconstructed clean signal $v_{1:T}$. The dashed lines and the numbers show the most-likely state segmentation $\arg\max_{s_{1:T}} p(s_{1:T}|\tilde{v}_{1:T})$.

CHAPTER 25

Switching Linear Dynamical Systems

HMMs assume that the underlying process is discrete. Linear dynamical systems that the underlying process is continuous. However, there are scenarios in which the underlying system might jump from one continuous regime to another. In this chapter we discuss a class of models that can be used in this situation. Unfortunately the technical demands of this class of models is somewhat more involved than in previous chapters, although the models are correspondingly more powerful.

25.1 Introduction

Complex timeseries which are not well described globally by a single Linear Dynamical System may be divided into segments, each modelled by a potentially different LDS. Such models can handle situations in which the underlying model 'jumps' from one parameter setting to another. For example a single LDS might well represent the normal flows in a chemical plant. When a break in a pipeline occurs, the dynamics of the system changes from one set of linear flow equations to another. This scenario can be modelled suing a sets of two linear systems, each with different parameters. The discrete latent variable at each time $s_t \in \{\text{normal}, \text{pipe broken}\}$ indicates which of the LDSs is most appropriate at the current time. This is called a Switching LDS and used in many disciplines, from econometrics to machine learning [12, 113, 177, 164, 163, 60, 57, 220, 309, 178].

25.2 The Switching LDS

At each time t, a switch variable $s_t \in 1, \ldots, S$ describes which of a set of LDSs is to be used. The observation (or 'visible') variable $\mathbf{v}_t \in \mathcal{R}^V$ is linearly related to the hidden state $\mathbf{h}_t \in \mathcal{R}^H$ by

$$\mathbf{v}_t = \mathbf{B}(s_t)\mathbf{h}_t + \boldsymbol{\eta}^v(s_t), \qquad \boldsymbol{\eta}^v(s_t) \sim \mathcal{N}\left(\boldsymbol{\eta}^v(s_t)|\bar{\mathbf{v}}(s_t), \boldsymbol{\Sigma}^v(s_t)\right) \tag{25.2.1}$$

Here s_t describes which of the set of emission matrices $\mathbf{B}(1), \ldots, \mathbf{B}(S)$ is active at time t. The observation noise $\boldsymbol{\eta}^v(s_t)$ is drawn from a Gaussian with mean $\bar{\mathbf{v}}(s_t)$ and covariance $\boldsymbol{\Sigma}^v(s_t)$. The transition dynamics of the continuous hidden state $\mathbf{h}_t$ is linear,

$$\mathbf{h}_t = \mathbf{A}(s_t)\mathbf{h}_{t-1} + \boldsymbol{\eta}^h(s_t), \qquad \boldsymbol{\eta}^h(s_t) \sim \mathcal{N}\left(\boldsymbol{\eta}^h(s_t)|\bar{\mathbf{h}}(s_t), \boldsymbol{\Sigma}^h(s_t)\right) \tag{25.2.2}$$

and the switch variable s_t selects a single transition matrix from the available set $\mathbf{A}(1), \ldots, \mathbf{A}(S)$. The Gaussian transition noise $\boldsymbol{\eta}^h(s_t)$ also depends on the switch variable. The dynamics of s_t itself is Markovian, with transition $p(s_t|s_{t-1})$. For the more general 'augmented' aSLDS model the switch s_t is dependent on both the previous s_{t-1} and $\mathbf{h}_{t-1}$. The model defines a joint distribution (see fig(25.1))

$$p(\mathbf{v}_{1:T}, \mathbf{h}_{1:T}, s_{1:T}) = \prod_{t=1}^{T} p(\mathbf{v}_t|\mathbf{h}_t, s_t)p(\mathbf{h}_t|\mathbf{h}_{t-1}, s_t)p(s_t|\mathbf{h}_{t-1}, s_{t-1})$$

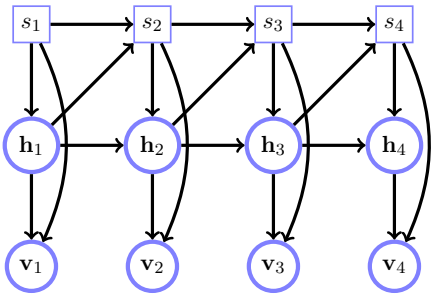

Figure 25.1: The independence structure of the aSLDS. Square nodes s_t denote discrete switch variables; $\mathbf{h}_t$ are continuous latent/hidden variables, and $\mathbf{v}_t$ continuous observed/visible variables. The discrete state s_t determines which Linear Dynamical system from a finite set of Linear Dynamical systems is operational at time t. In the SLDS links from h to s are not normally considered.

with

$$p(\mathbf{v}_t|\mathbf{h}_t, s_t) = \mathcal{N}\left(\mathbf{v}_t|\bar{\mathbf{v}}(s_t) + \mathbf{B}(s_t)\mathbf{h}_t, \mathbf{\Sigma}^v(s_t)\right), \qquad p(\mathbf{h}_t|\mathbf{h}_{t-1}, s_t) = \mathcal{N}\left(\mathbf{h}_t|\bar{\mathbf{h}}(s_t) + \mathbf{A}(s_t)\mathbf{h}_t, \mathbf{\Sigma}^h(s_t)\right) \tag{25.2.3}$$

At time $t = 1$, $p(s_1|\mathbf{h}_0, s_0)$ denotes the prior $p(s_1)$, and $p(\mathbf{h}_1|\mathbf{h}_0, s_1)$ denotes $p(\mathbf{h}_1|s_1)$.

The SLDS can be thought of as a marriage between a hidden Markov model and a Linear Dynamical System. The SLDS is also called a Jump Markov model/process, switching Kalman Filter, Switching Linear Gaussian State Space model, Conditional Linear Gaussian Model.

25.2.1 Exact inference is computationally intractable

Both exact filtered and smoothed inference in the SLDS is intractable, scaling exponentially with time. As an informal explanation, consider filtered posterior inference, for which, by analogy with equation (23.2.9), the forward pass is

$$p(s_{t+1}, \mathbf{h}_{t+1}|\mathbf{v}_{1:t+1}) = \sum_{s_t} \int_{\mathbf{h}_t} p(s_{t+1}, \mathbf{h}_{t+1}|s_t, \mathbf{h}_t, \mathbf{v}_{t+1}) p(s_t, \mathbf{h}_t|\mathbf{v}_{1:t}) \tag{25.2.4}$$

At timestep 1, $p(s_1, \mathbf{h}_1|\mathbf{v}_1) = p(\mathbf{h}_1|s_1, \mathbf{v}_1)p(s_1|\mathbf{v}_1)$ is an indexed set of Gaussians. At timestep 2, due to the summation over the states s_1, $p(s_2, \mathbf{h}_2|\mathbf{v}_{1:2})$ will be an indexed set of S Gaussians; similarly at timestep 3, it will be S^2 and, in general, gives rise to S^{t-1} Gaussians at time t. Even for small t, the number of components required to exactly represent the filtered distribution is therefore computationally intractable. Analogously, smoothing is also intractable. The origin of the intractability of the SLDS differs from 'structural intractability' that we've previously encountered. In the SLDS, in terms of the cluster variables $x_{1:T}$, with $x_t \equiv (s_t, \mathbf{h}_t)$ and visible variables $\mathbf{v}_{1:T}$, the graph of the distribution is singly-connected. From a purely graph theoretic viewpoint, one would therefore envisage little difficulty in carrying out inference. Indeed, as we saw above, the derivation of the filtering algorithm is straightforward since the graph is singly-connected. However, the numerical implementation of the algorithm is intractable since the description of the messages requires an exponentially increasing number of terms.

In order to deal with this intractability, several approximation schemes have been introduced [100, 113, 177, 164, 163]. Here we focus on techniques which approximate the switch conditional posteriors using a limited mixture of Gaussians. Since the exact posterior distributions are mixtures of Gaussians, albeit with an exponentially large number of components, the aim is to drop low weight components such that the resulting approximation accurately represents the posterior.

25.3 Gaussian Sum Filtering

Equation(25.2.4) describes the exact filtering recursion with an exponentially increasing number of components with time. In general, the influence of ancient observations will be much less relevant than that of recent observations. This suggests that the 'effective time' is limited and that therefore a corresponding limited number of components in the Gaussian mixture should suffice to accurately represent the filtered posterior[6].

Our aim is to form a recursion for $p(s_t, \mathbf{h}_t|\mathbf{v}_{1:t})$ based on a Gaussian mixture approximation of $p(\mathbf{h}_t|s_t, \mathbf{v}_{1:t})$. Given an approximation of the filtered distribution $p(s_t, \mathbf{h}_t|\mathbf{v}_{1:t}) \approx q(s_t, \mathbf{h}_t|\mathbf{v}_{1:t})$, the exact recursion equation (25.2.4) is approximated by

$$q(s_{t+1}, \mathbf{h}_{t+1}|\mathbf{v}_{1:t+1}) = \sum_{s_t} \int_{\mathbf{h}_t} p(s_{t+1}, \mathbf{h}_{t+1}|s_t, \mathbf{h}_t, \mathbf{v}_{t+1}) q(s_t, \mathbf{h}_t|\mathbf{v}_{1:t}) \tag{25.3.1}$$

This approximation to the filtered posterior at the next timestep will contain S times more components than in the previous timestep and, to prevent an exponential explosion in mixture components, we will need to subsequently collapse the mixture $q(s_{t+1}, \mathbf{h}_{t+1}|\mathbf{v}_{1:t+1})$ in a suitable way. We will deal with this issue once $q(s_{t+1}, \mathbf{h}_{t+1}|\mathbf{v}_{1:t+1})$ has been computed.

To derive the updates it is useful to break the new filtered approximation from equation (25.2.4) into continuous and discrete parts:

$$q(\mathbf{h}_t, s_t|\mathbf{v}_{1:t}) = q(\mathbf{h}_t|s_t, \mathbf{v}_{1:t}) q(s_t|\mathbf{v}_{1:t}) \tag{25.3.2}$$

and derive separate filtered update formulae, as described below.

25.3.1 Continuous filtering

The exact representation of $p(\mathbf{h}_t|s_t, \mathbf{v}_{1:t})$ is a mixture with $O\left(S^{t-1}\right)$ components. To retain computational feasibility we therefore approximate this with a limited I-component mixture

$$q(\mathbf{h}_t|s_t, \mathbf{v}_{1:t}) = \sum_{i_t=1}^{I} q(\mathbf{h}_t|i_t, s_t, \mathbf{v}_{1:t}) q(i_t|s_t, \mathbf{v}_{1:t}) \tag{25.3.3}$$

where $q(\mathbf{h}_t|i_t, s_t, \mathbf{v}_{1:t})$ is a Gaussian parameterised with mean $\mathbf{f}(i_t, s_t)$ and covariance $\mathbf{F}(i_t, s_t)$. Strictly speaking, we should use the notation $\mathbf{f}_t(i_t, s_t)$ since, for each time t, we have a set of means indexed by i_t, s_t, although we drop these dependencies in the notation used here.

An important remark is that many techniques approximate $p(\mathbf{h}_t|s_t, \mathbf{v}_{1:t})$ using a *single* Gaussian. Naturally, this gives rise to a mixture of Gaussians for $p(\mathbf{h}_t|\mathbf{v}_{1:t}) = \sum_{s_t} p(\mathbf{h}_t|s_t, \mathbf{v}_{1:t}) p(s_t|\mathbf{v}_{1:t})$. However, in making a single Gaussian approximation to $p(\mathbf{h}_t|s_t, \mathbf{v}_{1:t})$ the representation of the posterior may be poor. Our aim here is to maintain an accurate approximation to $p(\mathbf{h}_t|s_t, \mathbf{v}_{1:t})$ by using a *mixture* of Gaussians.

To find a recursion for the approximating distribution we first assume that we know the filtered approximation $q(\mathbf{h}_t, s_t|\mathbf{v}_{1:t})$ and then propagate this forwards using the exact dynamics. To do so consider first the relation

$$\begin{aligned} q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:t+1}) &= \sum_{s_t, i_t} q(\mathbf{h}_{t+1}, s_t, i_t|s_{t+1}, \mathbf{v}_{1:t+1}) \\ &= \sum_{s_t, i_t} q(\mathbf{h}_{t+1}|s_t, i_t, s_{t+1}, \mathbf{v}_{1:t+1}) q(s_t, i_t|s_{t+1}, \mathbf{v}_{1:t+1}) \end{aligned} \tag{25.3.4}$$

Wherever possible we now substitute the exact dynamics and evaluate each of the two factors above. The usefulness of decomposing the update in this way is that the new filtered approximation is of the form of a Gaussian mixture, where $q(\mathbf{h}_{t+1}|s_t, i_t, s_{t+1}, \mathbf{v}_{1:t+1})$ is Gaussian and $q(s_t, i_t|s_{t+1}, \mathbf{v}_{1:t+1})$ are the weights or mixing proportions of the components. We describe below how to compute these terms explicitly. Equation(25.3.4) produces a new Gaussian mixture with $I \times S$ components which we will collapse back to I components at the end of the computation.

Evaluating $q(\mathbf{h}_{t+1}|s_t, i_t, s_{t+1}, \mathbf{v}_{1:t+1})$

We aim to find a filtering recursion for $q(\mathbf{h}_{t+1}|s_t, i_t, s_{t+1}, \mathbf{v}_{1:t+1})$. Since this is conditional on switch states and components, this corresponds to a single LDS forward step which can be evaluated by considering first the joint distribution

$$q(\mathbf{h}_{t+1}, \mathbf{v}_{t+1}|s_t, i_t, s_{t+1}, \mathbf{v}_{1:t}) = \int_{\mathbf{h}_t} p(\mathbf{h}_{t+1}, \mathbf{v}_{t+1}|\mathbf{h}_t, \cancel{s_t}, \cancel{i_t}, s_{t+1}, \mathbf{v}_{1:t}) q(\mathbf{h}_t|s_t, i_t, \cancel{s_{t+1}}, \mathbf{v}_{1:t}) \tag{25.3.5}$$

and subsequently conditioning on $\mathbf{v}_{t+1}$. In the above we used the exact dynamics where possible.
Equation(25.3.5) states that we know the filtered information up to time t, in addition to knowing the switch states s_t, s_{t+1} and the mixture component index i_t. To ease the burden on notation we derive this for $\bar{\mathbf{h}}_t, \bar{\mathbf{v}}_t \equiv 0$ for all t. The exact forward dynamics is then given by

$$\mathbf{h}_{t+1} = \mathbf{A}(s_{t+1})\mathbf{h}_t + \boldsymbol{\eta}^h(s_{t+1}), \qquad \mathbf{v}_{t+1} = \mathbf{B}(s_{t+1})\mathbf{h}_{t+1} + \boldsymbol{\eta}^v(s_{t+1}), \tag{25.3.6}$$

Given the mixture component index i_t,

$$q(\mathbf{h}_t|\mathbf{v}_{1:t}, i_t, s_t) = \mathcal{N}\left(\mathbf{h}_t|\mathbf{f}(i_t, s_t), \mathbf{F}(i_t, s_t)\right) \tag{25.3.7}$$

we propagate this Gaussian with the exact dynamics equation (25.3.6). Then $q(\mathbf{h}_{t+1}, \mathbf{v}_{t+1}|s_t, i_t, s_{t+1}, \mathbf{v}_{1:t})$ is a Gaussian with covariance and mean elements

$$\begin{aligned}
&\boldsymbol{\Sigma}_{hh} = \mathbf{A}(s_{t+1})\mathbf{F}(i_t, s_t)\mathbf{A}^\mathsf{T}(s_{t+1}) + \boldsymbol{\Sigma}^h(s_{t+1}), \quad \boldsymbol{\Sigma}_{vv} = \mathbf{B}(s_{t+1})\boldsymbol{\Sigma}_{hh}\mathbf{B}^\mathsf{T}(s_{t+1}) + \boldsymbol{\Sigma}^v(s_{t+1}) \\
&\boldsymbol{\Sigma}_{vh} = \mathbf{B}(s_{t+1})\boldsymbol{\Sigma}_{hh} = \boldsymbol{\Sigma}_{hv}^\mathsf{T}, \quad \boldsymbol{\mu}_v = \mathbf{B}(s_{t+1})\mathbf{A}(s_{t+1})\mathbf{f}(i_t, s_t), \quad \boldsymbol{\mu}_h = \mathbf{A}(s_{t+1})\mathbf{f}(i_t, s_t)
\end{aligned} \tag{25.3.8}$$

These results are obtained from integrating the forward dynamics, equations (25.2.1,25.2.2) over $\mathbf{h}_t$, using result(8.2).

To find $q(\mathbf{h}_{t+1}|s_t, i_t, s_{t+1}, \mathbf{v}_{1:t+1})$ we condition $q(\mathbf{h}_{t+1}, \mathbf{v}_{t+1}|s_t, i_t, s_{t+1}, \mathbf{v}_{1:t})$ on $\mathbf{v}_{t+1}$ using the standard Gaussian conditioning formulae, result(8.3), to obtain

$$q(\mathbf{h}_{t+1}|s_t, i_t, s_{t+1}, \mathbf{v}_{1:t+1}) = \mathcal{N}\left(\mathbf{h}_{t+1}|\boldsymbol{\mu}_{h|v}, \boldsymbol{\Sigma}_{h|v}\right) \tag{25.3.9}$$

with

$$\boldsymbol{\mu}_{h|v} = \boldsymbol{\mu}_h + \boldsymbol{\Sigma}_{hv}\boldsymbol{\Sigma}_{vv}^{-1}\left(\mathbf{v}_{t+1} - \boldsymbol{\mu}_v\right), \qquad \boldsymbol{\Sigma}_{h|v} = \boldsymbol{\Sigma}_{hh} - \boldsymbol{\Sigma}_{hv}\boldsymbol{\Sigma}_{vv}^{-1}\boldsymbol{\Sigma}_{vh} \tag{25.3.10}$$

where the quantities required are defined in equation (25.3.8).

Evaluating the mixture weights $q(s_t, i_t|s_{t+1}, \mathbf{v}_{1:t+1})$

Up to a normalisation constant the mixture weight in equation (25.3.4) can be found from

$$q(s_t, i_t|s_{t+1}, \mathbf{v}_{1:t+1}) \propto q(\mathbf{v}_{t+1}|i_t, s_t, s_{t+1}, \mathbf{v}_{1:t})q(s_{t+1}|i_t, s_t, \mathbf{v}_{1:t})q(i_t|s_t, \mathbf{v}_{1:t})q(s_t|\mathbf{v}_{1:t}) \tag{25.3.11}$$

The first factor in equation (25.3.11), $q(\mathbf{v}_{t+1}|i_t, s_t, s_{t+1}, \mathbf{v}_{1:t})$ is Gaussian with mean $\boldsymbol{\mu}_v$ and covariance $\boldsymbol{\Sigma}_{vv}$, as given in equation (25.3.8). The last two factors $q(i_t|s_t, \mathbf{v}_{1:t})$ and $q(s_t|\mathbf{v}_{1:t})$ are given from the previous filtered iteration. Finally, $q(s_{t+1}|i_t, s_t, \mathbf{v}_{1:t})$ is found from

$$q(s_{t+1}|i_t, s_t, \mathbf{v}_{1:t}) = \begin{cases} \langle p(s_{t+1}|\mathbf{h}_t, s_t)\rangle_{q(\mathbf{h}_t|i_t, s_t, \mathbf{v}_{1:t})} & \text{augmented SLDS} \\ p(s_{t+1}|s_t) & \text{standard SLDS} \end{cases} \tag{25.3.12}$$

where the result above for the standard SLDS follows from the independence assumptions present in the standard SLDS. In the aSLDS, the term in equation (25.3.12) will generally need to be computed numerically. A simple approximation is to evaluate equation (25.3.12) at the mean value of the distribution $q(\mathbf{h}_t|i_t, s_t, \mathbf{v}_{1:t})$. To take covariance information into account an alternative would be to draw samples from the Gaussian $q(\mathbf{h}_t|i_t, s_t, \mathbf{v}_{1:t})$ and thus approximate the average of $p(s_{t+1}|\mathbf{h}_t, s_t)$ by sampling. Note that this does not equate Gaussian Sum filtering with a sequential sampling procedure, such as Particle Filtering, section(27.6.2). The sampling here is exact, for which no convergence issues arise.

Closing the recursion

We are now in a position to calculate equation (25.3.4). For each setting of the variable s_{t+1}, we have a mixture of $I \times S$ Gaussians. To prevent the number of components increasing exponentially with time, we numerically collapse $q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:t+1})$ back to I Gaussians to form

$$q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:t+1}) \to \sum_{i_{t+1}=1}^{I} q(\mathbf{h}_{t+1}|i_{t+1}, s_{t+1}, \mathbf{v}_{1:t+1})q(i_{t+1}|s_{t+1}, \mathbf{v}_{1:t+1}) \tag{25.3.13}$$

Any method of choice may be supplied to collapse a mixture to a smaller mixture. A straightforward approach is to repeatedly merge low-weight components, as explained in section(25.3.4). In this way the new mixture coefficients $q(i_{t+1}|s_{t+1}, \mathbf{v}_{1:t+1})$, $i_{t+1} \in 1, \ldots, I$ are defined. This completes the description of how to form a recursion for the continuous filtered posterior approximation $q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:t+1})$ in equation (25.3.2).

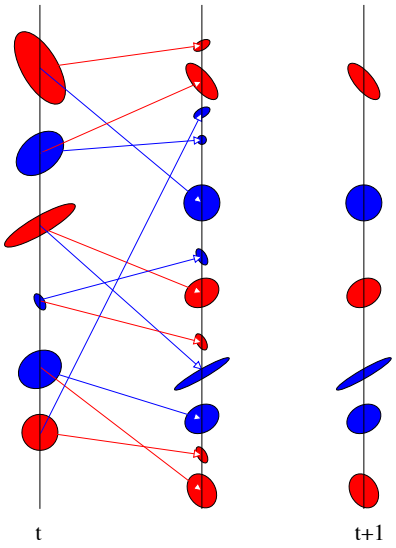

Figure 25.2: Gaussian Sum Filtering. The leftmost column depicts the previous Gaussian mixture approximation $q(\mathbf{h}_t, i_t|\mathbf{v}_{1:t})$ for two states $S = 2$ (red and blue) and three mixture components $I = 3$, with the mixture weight represented by the area of each oval. There are $S = 2$ different linear systems which take each of the components of the mixture into a new filtered state, the colour of the arrow indicating which dynamic system is used. After one time-step each mixture component branches into a further S components so that the joint approximation $q(\mathbf{h}_{t+1}, s_{t+1}|\mathbf{v}_{1:t+1})$ contains S^2I components (middle column). To keep the representation computationally tractable the mixture of Gaussians for each state s_{t+1} is collapsed back to I components. This means that each coloured state needs to be approximated by a smaller I component mixture of Gaussians. There are many ways to achieve this. A naive but computationally efficient approach is to simply ignore the lowest weight components, as depicted on the right column, see `mix2mix.m`.

25.3.2 Discrete filtering

A recursion for the switch variable distribution in equation (25.3.2) is

$$q(s_{t+1}|\mathbf{v}_{1:t+1}) \propto \sum_{i_t,s_t} q(s_{t+1}, i_t, s_t, \mathbf{v}_{t+1}, \mathbf{v}_{1:t}) \tag{25.3.14}$$

The r.h.s. of the above equation is proportional to

$$\sum_{s_t,i_t} q(\mathbf{v}_{t+1}|s_{t+1}, i_t, s_t, \mathbf{v}_{1:t})q(s_{t+1}|i_t, s_t, \mathbf{v}_{1:t})q(i_t|s_t, \mathbf{v}_{1:t})q(s_t|\mathbf{v}_{1:t}) \tag{25.3.15}$$

for which all terms have been computed during the recursion for $q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:t+1})$. We now have all the quantities required to compute the Gaussian Sum approximation of the filtering forward pass. A schematic representation of Gaussian Sum filtering is given in fig(25.2) and the pseudo code is presented in algorithm(25.1). See also `SLDSforward.m`.

25.3.3 The likelihood $p(\mathbf{v}_{1:T})$

The likelihood $p(\mathbf{v}_{1:T})$ may be found from

$$p(\mathbf{v}_{1:T}) = \prod_{t=0}^{T-1} p(\mathbf{v}_{t+1}|\mathbf{v}_{1:t}) \tag{25.3.16}$$

where

$$p(\mathbf{v}_{t+1}|\mathbf{v}_{1:t}) \approx \sum_{i_t,s_t,s_{t+1}} q(\mathbf{v}_{t+1}|i_t, s_t, s_{t+1}, \mathbf{v}_{1:t})q(s_{t+1}|i_t, s_t, \mathbf{v}_{1:t})q(i_t|s_t, \mathbf{v}_{1:t})q(s_t|\mathbf{v}_{1:t})$$

In the above expression, all terms have been computed in forming the recursion for the filtered posterior $q(\mathbf{h}_{t+1}, s_{t+1}|\mathbf{v}_{1:t+1})$.

25.3.4 Collapsing Gaussians

Given a mixture of N Gaussians

$$p(\mathbf{x}) = \sum_{i=1}^{N} p_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{25.3.17}$$

we wish to collapse this to a smaller $K < N$ mixture of Gaussians. We describe a simple method which has the advantage of computational efficiency, but the disadvantage that no spatial information about the

Algorithm 25.1 aSLDS Forward Pass. Approximate the filtered posterior $p(s_t|\mathbf{v}_{1:t}) \equiv \alpha_t$, $p(\mathbf{h}_t|s_t, \mathbf{v}_{1:t}) \equiv \sum_{i_t} w_t(i_t, s_t)\mathcal{N}\left(\mathbf{h}_t|\mathbf{f}_t(i_t, s_t), \mathbf{F}_t(i_t, s_t)\right)$. Also return the approximate log-likelihood $L \equiv \log p(\mathbf{v}_{1:T})$. I_t are the number of components in each Gaussian mixture approximation. We require $I_1 = 1, I_2 \leq S, I_t \leq S \times I_{t-1}$. $\theta(s) = \mathbf{A}(s), \mathbf{B}(s), \mathbf{\Sigma}^h(s), \mathbf{\Sigma}^v(s), \bar{\mathbf{h}}(s), \bar{\mathbf{v}}(s)$.

for $s_1 \leftarrow 1$ **to** S **do**
  $\{\mathbf{f}_1(1, s_1), \mathbf{F}_1(1, s_1), \hat{p}\} = \text{LDSFORWARD}(0, 0, \mathbf{v}_1; \theta(s_1))$
  $\alpha_1 \leftarrow p(s_1)\hat{p}$
end for

for $t \leftarrow 2$ **to** T **do**
  for $s_t \leftarrow 1$ **to** S **do**
    for $i \leftarrow 1$ **to** I_{t-1}, **and** $s \leftarrow 1$ **to** S **do**
      $\{\boldsymbol{\mu}_{x|y}(i, s), \mathbf{\Sigma}_{x|y}(i, s), \hat{p}\} = \text{LDSFORWARD}(\mathbf{f}_{t-1}(i, s), \mathbf{F}_{t-1}(i, s), \mathbf{v}_t; \theta(s_t))$
      $p^*(s_t|i, s) \equiv \langle p(s_t|\mathbf{h}_{t-1}, s_{t-1} = s)\rangle_{p(\mathbf{h}_{t-1}|i_{t-1}=i, s_{t-1}=s, \mathbf{v}_{1:t-1})}$
      $p'(s_t, i, s) \leftarrow w_{t-1}(i, s)p^*(s_t|i, s)\alpha_{t-1}(s)\hat{p}$
    end for
    Collapse the $I_{t-1} \times S$ mixture of Gaussians defined by $\boldsymbol{\mu}_{x|y}$,$\mathbf{\Sigma}_{x|y}$, and weights $p(i, s|s_t) \propto p'(s_t, i, s)$ to a Gaussian with I_t components, $p(\mathbf{h}_t|s_t, \mathbf{v}_{1:t}) \approx \sum_{i_t=1}^{I_t} p(i_t|s_t, \mathbf{v}_{1:t})p(\mathbf{h}_t|s_t, i_t, \mathbf{v}_{1:t})$. This defines the new means $\mathbf{f}_t(i_t, s_t)$, covariances $\mathbf{F}_t(i_t, s_t)$ and mixture weights $w_t(i_t, s_t) \equiv p(i_t|s_t, \mathbf{v}_{1:t})$.
    Compute $\alpha_t(s_t) \propto \sum_{i,s} p'(s_t, i, s)$
  end for
  normalise α_t
  $L \leftarrow L + \log \sum_{s_t,i,s} p'(s_t, i, s)$
end for

mixture is used[281]. First we describe how to collapse a mixture to a single Gaussian. This can be achieved by finding the mean and covariance of the mixture distribution (25.3.17). These are

$$\boldsymbol{\mu} = \sum_i p_i \boldsymbol{\mu}_i, \qquad \mathbf{\Sigma} = \sum_i p_i \left(\mathbf{\Sigma}_i + \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\mathsf{T}\right) - \boldsymbol{\mu}\boldsymbol{\mu}^\mathsf{T} \tag{25.3.18}$$

To collapse a mixture then to a K-component mixture we may first retain the $K - 1$ Gaussians with the largest mixture weights. The remaining $N - K + 1$ Gaussians are simply merged to a single Gaussian using the above method. Alternative heuristics such as recursively merging the two Gaussians with the lowest mixture weights are also reasonable.

More sophisticated methods which retain some spatial information would clearly be potentially useful. The method presented in [177] is a suitable approach which considers removing Gaussians which are spatially similar (and not just low-weight components), thereby retaining a sense of diversity over the possible solutions. In applications with many thousands of timesteps, speed can be a factor in determining which method of collapsing Gaussians is to be preferred.

25.3.5 Relation to other methods

Gaussian Sum Filtering can be considered a form of 'analytical particle filtering', section(27.6.2), in which instead of point distributions (delta functions) being propagated, Gaussians are propagated. The collapse operation to a smaller number of Gaussians is analogous to resampling in Particle Filtering. Since a Gaussian is more expressive than a delta-function, the Gaussian Sum filter is generally an improved approximation technique over using point particles. See [17] for a numerical comparison.

25.4 Gaussian Sum Smoothing

Approximating the smoothed posterior $p(\mathbf{h}_t, s_t|\mathbf{v}_{1:T})$ is more involved than filtering and requires additional approximations. For this reason smoothing is more prone to failure since there are more assumptions that need to be satisfied for the approximations to hold. The route we take here is to assume that a Gaussian Sum filtered approximation has been carried out, and then approximate the γ backward pass, analogous to that

of section(23.2.4). By analogy with the RTS smoothing recursion equation (23.2.20), the exact backward pass for the SLDS reads

$$p(\mathbf{h}_t, s_t|\mathbf{v}_{1:T}) = \sum_{s_{t+1}} \int_{\mathbf{h}_{t+1}} p(\mathbf{h}_t, s_t|\mathbf{h}_{t+1}, s_{t+1}, \mathbf{v}_{1:t}) p(\mathbf{h}_{t+1}, s_{t+1}|\mathbf{v}_{1:T}) \tag{25.4.1}$$

where $p(\mathbf{h}_{t+1}, s_{t+1}|\mathbf{v}_{1:T}) = p(s_{t+1}|\mathbf{v}_{1:T}) p(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T})$ is composed of the discrete and continuous components of the smoothed posterior at the next time step. The recursion runs backwards in time, beginning with the initialisation $p(\mathbf{h}_T, s_T|\mathbf{v}_{1:T})$ set by the filtered result (at time $t = T$, the filtered and smoothed posteriors coincide). Apart from the fact that the number of mixture components will increase at each step, computing the integral over $\mathbf{h}_{t+1}$ in equation (25.4.1) is problematic since the conditional distribution term is non-Gaussian in $\mathbf{h}_{t+1}$. For this reason it is more useful derive an approximate recursion by beginning with the exact relation

$$p(s_t, \mathbf{h}_t|\mathbf{v}_{1:T}) = \sum_{s_{t+1}} p(s_{t+1}|\mathbf{v}_{1:T}) p(\mathbf{h}_t|s_t, s_{t+1}, \mathbf{v}_{1:T}) p(s_t|s_{t+1}, \mathbf{v}_{1:T}) \tag{25.4.2}$$

which can be expressed more directly in terms of the SLDS dynamics as

$$p(s_t, \mathbf{h}_t|\mathbf{v}_{1:T}) = \sum_{s_{t+1}} p(s_{t+1}|\mathbf{v}_{1:T}) \left\langle p(\mathbf{h}_t|\mathbf{h}_{t+1}, s_t, s_{t+1}, \mathbf{v}_{1:t}, \cancel{\mathbf{v}_{t+1:T}}) \right\rangle_{p(\mathbf{h}_{t+1}|s_t, s_{t+1}, \mathbf{v}_{1:T})}$$
$$\times \left\langle p(s_t|\mathbf{h}_{t+1}, s_{t+1}, \mathbf{v}_{1:T}) \right\rangle_{p(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T})} \tag{25.4.3}$$

In forming the recursion we assume access to the distribution $p(s_{t+1}, \mathbf{h}_{t+1}|\mathbf{v}_{1:T})$ from the future timestep. However, we also require the distribution $p(\mathbf{h}_{t+1}|s_t, s_{t+1}, \mathbf{v}_{1:T})$ which is not directly known and needs to be inferred, in itself a computationally challenging task. In the *Expectation Correction* (EC) approach [17] one assumes the approximation (see fig(25.3))

$$p(\mathbf{h}_{t+1}|s_t, s_{t+1}, \mathbf{v}_{1:T}) \approx p(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T}) \tag{25.4.4}$$

resulting in an approximate recursion for the smoothed posterior,

$$p(s_t, \mathbf{h}_t|\mathbf{v}_{1:T}) \approx \sum_{s_{t+1}} p(s_{t+1}|\mathbf{v}_{1:T}) \left\langle p(\mathbf{h}_t|\mathbf{h}_{t+1}, s_t, s_{t+1}, \mathbf{v}_{1:t}) \right\rangle_{\mathbf{h}_{t+1}} \left\langle p(s_t|\mathbf{h}_{t+1}, s_{t+1}, \mathbf{v}_{1:T}) \right\rangle_{\mathbf{h}_{t+1}} \tag{25.4.5}$$

where $\langle \cdot \rangle_{\mathbf{h}_{t+1}}$ represents averaging with respect to the distribution $p(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T})$. In carrying out the approximate recursion, (25.4.5) we will end up with a mixture of Gaussians that grows at each timestep. To avoid the exponential explosion problem, we use a finite mixture approximation, $q(\mathbf{h}_{t+1}, s_{t+1}|\mathbf{v}_{1:T})$:

$$p(\mathbf{h}_{t+1}, s_{t+1}|\mathbf{v}_{1:T}) \approx q(\mathbf{h}_{t+1}, s_{t+1}|\mathbf{v}_{1:T}) = q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T}) q(s_{t+1}|\mathbf{v}_{1:T}) \tag{25.4.6}$$

and plug this into the approximate recursion above. From equation (25.4.5) a recursion for the approximation is given by

$$q(\mathbf{h}_t, s_t|\mathbf{v}_{1:T}) = \sum_{s_{t+1}} q(s_{t+1}|\mathbf{v}_{1:T}) \underbrace{\left\langle q(\mathbf{h}_t|\mathbf{h}_{t+1}, s_t, s_{t+1}, \mathbf{v}_{1:t}) \right\rangle_{q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T})}}_{q(\mathbf{h}_t|s_t, s_{t+1}, \mathbf{v}_{1:T})} \underbrace{\left\langle q(s_t|\mathbf{h}_{t+1}, s_{t+1}, \mathbf{v}_{1:t}) \right\rangle_{q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T})}}_{q(s_t|s_{t+1}, \mathbf{v}_{1:T})} \tag{25.4.7}$$

As for filtering, wherever possible, we replace approximate terms by their exact counterparts and parameterise the posterior using

$$q(\mathbf{h}_{t+1}, s_{t+1}|\mathbf{v}_{1:T}) = q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T}) q(s_{t+1}|\mathbf{v}_{1:T}) \tag{25.4.8}$$

To reduce the notational burden here we outline the method only for the case of using a single component approximation in both the forward and backward passes. The extension to using a mixture to approximate each $p(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T})$ is conceptually straightforward and deferred to section(25.4.4). In the single Gaussian case we assume we have a Gaussian approximation available for

$$q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T}) = \mathcal{N}\left(\mathbf{h}_{t+1}|\mathbf{g}(s_{t+1}), \mathbf{G}(s_{t+1})\right) \tag{25.4.9}$$

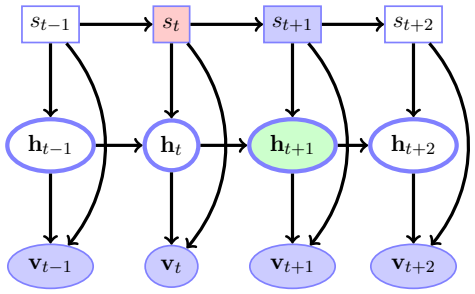

Figure 25.3: The EC backpass approximates $p(\mathbf{h}_{t+1}|s_{t+1}, s_t, \mathbf{v}_{1:T})$ by $p(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T})$. The motivation for this is that s_t influences $\mathbf{h}_{t+1}$ only indirectly through $\mathbf{h}_t$. However, $\mathbf{h}_t$ will most likely be heavily influenced by $\mathbf{v}_{1:t}$, so that not knowing the state of s_t is likely to be of secondary importance. The green shaded node is the variable we wish to find the posterior for. The values of the blue shaded nodes are known, and the red shaded node indicates a known variable which is assumed unknown in forming the approximation.

25.4.1 Continuous smoothing

For given s_t, s_{t+1}, an RTS style recursion for the smoothed continuous is obtained from equation (25.4.7), giving

$$q(\mathbf{h}_t|s_t, s_{t+1}, \mathbf{v}_{1:T}) = \int_{\mathbf{h}_{t+1}} p(\mathbf{h}_t|\mathbf{h}_{t+1}, s_t, s_{t+1}, \mathbf{v}_{1:t}) q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T}) \tag{25.4.10}$$

To compute equation (25.4.10) we then perform a single update of the LDS backward recursion, section(24.4.2).

25.4.2 Discrete smoothing

The second average in equation (25.4.7) corresponds to a recursion for the discrete variable and is given by

$$\langle q(s_t|\mathbf{h}_{t+1}, s_{t+1}, \mathbf{v}_{1:t})\rangle_{q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T})} \equiv q(s_t|s_{t+1}, \mathbf{v}_{1:T}). \tag{25.4.11}$$

The average of $q(s_t|\mathbf{h}_{t+1}, s_{t+1}, \mathbf{v}_{1:t})$ with respect to $q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T})$ cannot be achieved in closed form. A simple approach is to approximate the average by evaluation at the mean[1]

$$\langle q(s_t|\mathbf{h}_{t+1}, s_{t+1}\mathbf{v}_{1:t})\rangle_{q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T})} \approx q(s_t|\mathbf{h}_{t+1}, s_{t+1}, \mathbf{v}_{1:t})\big|_{\mathbf{h}_{t+1}=\langle \mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T}\rangle} \tag{25.4.12}$$

where $\langle \mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T}\rangle$ is the mean of $\mathbf{h}_{t+1}$ with respect to $q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T})$.

Replacing $\mathbf{h}_{t+1}$ by its mean gives the approximation

$$\langle q(s_t|\mathbf{h}_{t+1}, s_{t+1}, \mathbf{v}_{1:t})\rangle_{q(\mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T})} \approx \frac{1}{Z} \frac{e^{-\frac{1}{2} z_{t+1}^{\mathsf{T}}(s_t, s_{t+1}) \boldsymbol{\Sigma}^{-1}(s_t, s_{t+1}|\mathbf{v}_{1:t}) z_{t+1}(s_t, s_{t+1})}}{\sqrt{\det\left(\boldsymbol{\Sigma}(s_t, s_{t+1}|\mathbf{v}_{1:t})\right)}} q(s_t|s_{t+1}, \mathbf{v}_{1:t}) \tag{25.4.13}$$

where

$$z_{t+1}(s_t, s_{t+1}) \equiv \langle \mathbf{h}_{t+1}|s_{t+1}, \mathbf{v}_{1:T}\rangle - \langle \mathbf{h}_{t+1}|s_t, s_{t+1}, \mathbf{v}_{1:t}\rangle \tag{25.4.14}$$

and Z ensures normalisation over s_t. $\boldsymbol{\Sigma}(s_t, s_{t+1}|\mathbf{v}_{1:t})$ is the filtered covariance of $\mathbf{h}_{t+1}$ given s_t, s_{t+1} and the observations $\mathbf{v}_{1:t}$, which may be taken from $\boldsymbol{\Sigma}_{hh}$ in equation (25.3.8). Approximations which take covariance information into account can also be considered, although the above simple (and fast) method may suffice in practice [17, 195].

25.4.3 Collapsing the mixture

From section(25.4.1) and section(25.4.2) we now have all the terms in equation (25.4.8) to compute the approximation to equation (25.4.7). Due to the summatino over s_{t+1} in equation (25.4.7), the number of mixture components is multiplied by S at each iteration. To prevent an exponential explosion of components, the mixture equation (25.4.7) is then collapsed to a single Gaussian

$$q(\mathbf{h}_t, s_t|\mathbf{v}_{1:T}) \to q(\mathbf{h}_t|s_t, \mathbf{v}_{1:T}) q(s_t|\mathbf{v}_{1:T})) \tag{25.4.15}$$

The collapse to a mixture is discussed in section(25.4.4).

[1] In general this approximation has the form $\langle f(x)\rangle \approx f(\langle x\rangle)$.

Algorithm 25.2 aSLDS: EC Backward Pass. Approximates $p(s_t|\mathbf{v}_{1:T})$ and $p(\mathbf{h}_t|s_t, \mathbf{v}_{1:T}) \equiv \sum_{j_t=1}^{J_t} u_t(j_t, s_t)\mathcal{N}(\mathbf{g}_t(j_t, s_t), \mathbf{G}_t(j_t, s_t))$ using a mixture of Gaussians. $J_T = I_T, J_t \leq S \times I_t \times J_{t+1}$. This routine needs the results from algorithm(25.1).

$\mathbf{G}_T \leftarrow \mathbf{F}_T$, $\mathbf{g}_T \leftarrow \mathbf{f}_T$, $u_T \leftarrow w_T$
for $t \leftarrow T-1$ **to** 1 **do**
  for $s \leftarrow 1$ **to** S, $s' \leftarrow 1$ **to** S, $i \leftarrow 1$ **to** I_t, $j' \leftarrow 1$ **to** J_{t+1} **do**
    $(\boldsymbol{\mu}, \boldsymbol{\Sigma})(i, s, j', s') = \text{LDSBACKWARD}(\mathbf{g}_{t+1}(j', s'), \mathbf{G}_{t+1}(j', s'), \mathbf{f}_t(i, s), \mathbf{F}_t(i, s), \theta(s'))$
    $p(i_t, s_t|j_{t+1}, s_{t+1}, v_{1:T}) = \langle p(s_t = s, i_t = i|\mathbf{h}_{t+1}, s_{t+1} = s', j_{t+1} = j', \mathbf{v}_{1:t})\rangle_{p(\mathbf{h}_{t+1}|s_{t+1}=s', j_{t+1}=j', \mathbf{v}_{1:T})}$
    $p(i, s, j', s'|\mathbf{v}_{1:T}) \leftarrow p(s_{t+1} = s'|\mathbf{v}_{1:T})u_{t+1}(j', s')p(i_t, s_t|j_{t+1}, s_{t+1}, v_{1:T})$
  end for
  for $s_t \leftarrow 1$ **to** S **do**
    Collapse the mixture defined by weights $p(i_t = i, s_{t+1} = s', j_{t+1} = j'|s_t, \mathbf{v}_{1_T}) \propto p(i, s_t, j', s'|\mathbf{v}_{1:T})$, means $\boldsymbol{\mu}(i_t, s_t, j_{t+1}, s_{t+1})$ and covariances $\boldsymbol{\Sigma}(i_t, s_t, j_{t+1}, s_{t+1})$ to a mixture with J_t components. This defines the new means $\mathbf{g}_t(j_t, s_t)$, covariances $\mathbf{G}_t(j_t, s_t)$ and mixture weights $u_t(j_t, s_t)$.
    $p(s_t|\mathbf{v}_{1:T}) \leftarrow \sum_{i_t, j', s'} p(i_t, s_t, j', s'|\mathbf{v}_{1:T})$
  end for
end for

25.4.4 Using mixtures in smoothing

The extension to the mixture case is straightforward based on the representation

$$p(\mathbf{h}_t|s_t, \mathbf{v}_{1:T}) \approx \sum_{j_t=1}^{J} q(j_t|s_t, \mathbf{v}_{1:T})q(\mathbf{h}_t|s_t, j_t, \mathbf{v}_{1:T}). \tag{25.4.16}$$

Analogously to the case with a single component,

$$q(\mathbf{h}_t, s_t|\mathbf{v}_{1:T}) = \sum_{i_t, j_{t+1}, s_{t+1}} p(s_{t+1}|\mathbf{v}_{1:T})p(j_{t+1}|s_{t+1}, \mathbf{v}_{1:T})q(\mathbf{h}_t|j_{t+1}, s_{t+1}, i_t, s_t, \mathbf{v}_{1:T}) \times \langle q(i_t, s_t|\mathbf{h}_{t+1}, j_{t+1}, s_{t+1}, \mathbf{v}_{1:t})\rangle_{q(\mathbf{h}_{t+1}|j_{t+1}, s_{t+1}, \mathbf{v}_{1:T})} \tag{25.4.17}$$

The average in the last line of the above equation can be tackled using the same techniques as outlined in the single Gaussian case. To approximate $q(\mathbf{h}_t|j_{t+1}, s_{t+1}, i_t, s_t, \mathbf{v}_{1:T})$ we consider this as the marginal of the joint distribution

$$q(\mathbf{h}_t, \mathbf{h}_{t+1}|i_t, s_t, j_{t+1}, s_{t+1}, \mathbf{v}_{1:T}) = q(\mathbf{h}_t|\mathbf{h}_{t+1}, i_t, s_t, j_{t+1}, s_{t+1}, \mathbf{v}_{1:t})q(\mathbf{h}_{t+1}|i_t, s_t, j_{t+1}, s_{t+1}, \mathbf{v}_{1:T}) \tag{25.4.18}$$

As in the case of a single mixture, the problematic term is $q(\mathbf{h}_{t+1}|i_t, s_t, j_{t+1}, s_{t+1}, \mathbf{v}_{1:T})$. Analogously to equation (25.4.4), we make the assumption

$$q(\mathbf{h}_{t+1}|i_t, s_t, j_{t+1}, s_{t+1}, \mathbf{v}_{1:T}) \approx q(\mathbf{h}_{t+1}|j_{t+1}, s_{t+1}, \mathbf{v}_{1:T}) \tag{25.4.19}$$

meaning that information about the current switch state s_t, i_t is ignored. We can then form

$$p(\mathbf{h}_t|s_t, \mathbf{v}_{1:T}) = \sum_{i_t, j_{t+1}, s_{t+1}} p(i_t, j_{t+1}, s_{t+1}|s_t, \mathbf{v}_{1:T})p(\mathbf{h}_t|i_t, s_t, j_{t+1}, s_{t+1}, \mathbf{v}_{1:T}) \tag{25.4.20}$$

This mixture can then be collapsed to a smaller mixture using any method of choice, to give

$$p(\mathbf{h}_t|s_t, \mathbf{v}_{1:T}) \approx \sum_{j_t} q(j_t|s_t, \mathbf{v}_{1:T})q(\mathbf{h}_t|j_t, \mathbf{v}_{1:T}) \tag{25.4.21}$$

The resulting procedure is sketched in algorithm(25.2), including using mixtures in both the forward and backward passes.

25.4.5 Relation to other methods

A classical smoothing approximation for the SLDS is *generalised pseudo Bayes* (GPB) [12, 163, 162]. In GPB one starts from the exact recursion

$$p(s_t|\mathbf{v}_{1:T}) = \sum_{s_{t+1}} p(s_t, s_{t+1}|\mathbf{v}_{1:T}) = \sum_{s_{t+1}} p(s_t|s_{t+1}, \mathbf{v}_{1:T})p(s_{t+1}|\mathbf{v}_{1:T}) \tag{25.4.22}$$

The quantity $p(s_t|s_{t+1}, \mathbf{v}_{1:T})$ is difficult to obtain and GPB makes the approximation

$$p(s_t|s_{t+1}, \mathbf{v}_{1:T}) \approx p(s_t|s_{t+1}, \mathbf{v}_{1:t}) \tag{25.4.23}$$

Plugging this into equation (25.4.22) we have

$$p(s_t|\mathbf{v}_{1:T}) \approx \sum_{s_{t+1}} p(s_t|s_{t+1}, \mathbf{v}_{1:t})p(s_{t+1}|\mathbf{v}_{1:T}) \tag{25.4.24}$$

$$= \sum_{s_{t+1}} \frac{p(s_{t+1}|s_t)p(s_t|\mathbf{v}_{1:t})}{\sum_{s_t} p(s_{t+1}|s_t)p(s_t|\mathbf{v}_{1:t})} p(s_{t+1}|\mathbf{v}_{1:T}) \tag{25.4.25}$$

The recursion is initialised with the approximate filtered $p(s_T|\mathbf{v}_{1:T})$. Computing the smoothed recursion for the switch states in GPB is then equivalent to running the RTS backward pass on a hidden Markov model, independently of the backward recursion for the continuous variables. The only information the GPB method uses to form the smoothed distribution $p(s_t|\mathbf{v}_{1:T})$ from the filtered distribution $p(s_t|\mathbf{v}_{1:t})$ is the Markov switch transition $p(s_{t+1}|s_t)$. This approximation drops information from the future since information passed via the continuous variables is not taken into account. In contrast to GPB, the EC Gaussian smoothing technique preserves future information passing through the continuous variables. As for EC, GPB forms an approximation for $p(\mathbf{h}_t|s_t, \mathbf{v}_{1:T})$ by using the recursion (25.4.8) where $q(s_t|s_{t+1}, \mathbf{v}_{1:T})$ is given by $q(s_t|s_{t+1}, \mathbf{v}_{1:t})$. In `SLDSbackward.m` one may choose to use either EC or GBP.

Example 25.1 (Traffic Flow)**.** A illustration of modelling and inference with a SLDS is to consider a simple network of traffic flow, fig(25.4). Here there are 4 junctions a, b, c, d and traffic flows along the roads in the direction indicated. Traffic flows into the junction at a and then goes via different routes to d. Flow out of a junction must match the flow in to a junction (up to noise). There are traffic light switches at junctions a and b which, depending on their state, route traffic differently along the roads.

Using ϕ to denote the clean (noise free) flow, we model the flows using the switching linear system:

$$\left.\begin{array}{r} \phi_a(t) \\ \phi_{a\to d}(t) \\ \phi_{a\to b}(t) \\ \phi_{b\to d}(t) \\ \phi_{b\to c}(t) \\ \phi_{c\to d}(t) \end{array}\right\} = \left\{\begin{array}{l} \phi_a(t-1) \\ \phi_a(t-1)\left(0.75 \times \mathbb{I}\left[s_a(t)=1\right] + 1 \times \mathbb{I}\left[s_a(t)=2\right]\right) \\ \phi_a(t-1)\left(0.25 \times \mathbb{I}\left[s_a(t)=1\right] + 1 \times \mathbb{I}\left[s_a(t)=3\right]\right) \\ \phi_{a\to b}(t-1) 0.5 \times \mathbb{I}\left[s_b(t)=1\right] \\ \phi_{a\to b}(t-1)\left(0.5 \times \mathbb{I}\left[s_b(t)=1\right] + 1 \times \mathbb{I}\left[s_b(t)=2\right]\right) \\ \phi_{b\to c}(t-1) \end{array}\right. \tag{25.4.26}$$

By identifying the flows at time t with a 6 dimensional vector hidden variable $\mathbf{h}_t$, we can write the above flow equations as

$$\mathbf{h}_t = \mathbf{A}(s_t)\mathbf{h}_{t-1} + \boldsymbol{\eta}_t^h \tag{25.4.27}$$

for a set of suitably defined matrices $\mathbf{A}(s)$ indexed by the switch variable $s = s_a \bigotimes s_b$, which takes $3 \times 2 = 6$ states. We additionally include noise terms to model cars parking or de-parking during a single timestep. The covariance $\boldsymbol{\Sigma}^h$ is diagonal with a larger variance at the inflow point a to model that the total volume of traffic entering the system can vary.

Noisy measurements of the flow into the network are taken at a

$$v_{1,t} = \phi_a(t) + \eta_1^v(t) \tag{25.4.28}$$

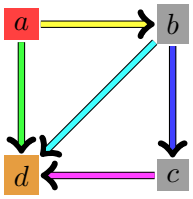

Figure 25.4: A representation of the traffic flow between junctions at a,b,c,d, with traffic lights at a and b. If $s_a = 1$ $a \to d$ and $a \to b$ carry 0.75 and 0.25 of the flow out of a respectively. If $s_a = 2$ all the flow from a goes through $a \to d$; for $s_a = 3$, all the flow goes through $a \to b$. For $s_b = 1$ the flow out of b is split equally between $b \to d$ and $b \to c$. For $s_b = 2$ all flow out of b goes along $b \to c$.

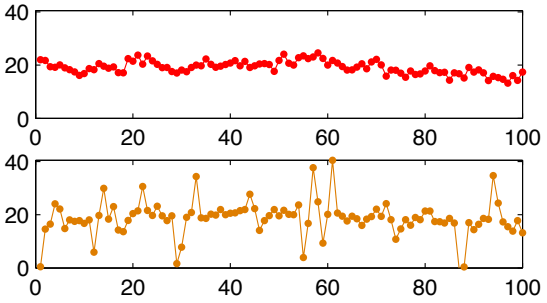

Figure 25.5: Time evolution of the traffic flow measured at two points in the network. Sensors measure the total flow into the network $\phi_a(t)$ and the total flow out of the network, $\phi_d(t) = \phi_{a\to d}(t) + \phi_{b\to d}(t) + \phi_{c\to d}(t)$. The total inflow at a undergoes a random walk. Note that the flow measured at d can momentarily drop to zero if all traffic is routed through $a \to b \to c$ in two consecutive time steps.

along with a noisy measurement of the total flow out of the system at d,

$$v_{2,t} = \phi_d(t) = \phi_{a\to d}(t) + \phi_{b\to d}(t) + \phi_{c\to d}(t) + \eta_2^v(t) \tag{25.4.29}$$

The observation model can be represented by $\mathbf{v}_t = \mathbf{B}\mathbf{h}_t + \boldsymbol{\eta}_t^v$ using a constant 2×6 projection matrix $\mathbf{B}$. The switch variables follow a simple Markov transition $p(s_t|s_{t-1})$ which biases the switches to remain in the same state in preference to jumping to another state. See `demoSLDStraffic.m` for details.

Given the above system and a prior which initialises all flow at a, we draw samples from the model using forward (ancestral) sampling which form the observations $\mathbf{v}_{1:100}$, fig(25.5). Using only the observations and the known model structure we then attempt to infer the latent switch variables and traffic flows using Gaussian Sum filtering and smoothing (EC method) with 2 mixture components per switch state, fig(25.6).

We note that a naive HMM approximation based on discretising each continuous flow into 20 bins would contain $2 \times 3 \times 20^6$ or 384 million states. Even for modest size problems, a naive approximation based on discretisation is therefore impractical.

Example 25.2 (Following the price trend)**.** The following is a simple model of the price trend of a stock, which assumes that the price tends to continue going up (or down) for a while before it reverses direction:

$$h_1(t) = h_1(t-1) + h_2(t-1) + \eta_1^h(s_t) \tag{25.4.30}$$
$$h_2(t) = \mathbb{I}\left[s(t) = 1\right] h_2(t-1) + \eta_2^h(s_t) \tag{25.4.31}$$
$$v(t) = h_1(t) + \eta^v(s_t) \tag{25.4.32}$$

here h_1 represents the price and h_2 the direction. There is only a single observation variable at each time, which is the price plus a small amount of noise. There are two switch states, $\mathrm{dom}(s_t) = \{1, 2\}$. When $s_t = 1$, the model functions normally, with the direction being equal to the previous direction plus a small amount of noise $\eta_2^h(s_t = 1)$. When $s_t = 2$ however, the direction is sampled from a Gaussian with a large variance. The transition $p(s_t|s_{t-1})$ is set so that normal dynamics is more likely, and when $s_t = 2$ it is likely to go back to normal dynamics the next timestep. Full details are in `SLDSpricemodel.mat`. In fig(25.7) we plot some samples from the model and also smoothed inference of the switch distribution, showing how we can a posteriori infer the likely changes in the stock price direction. See also exercise(25.1).

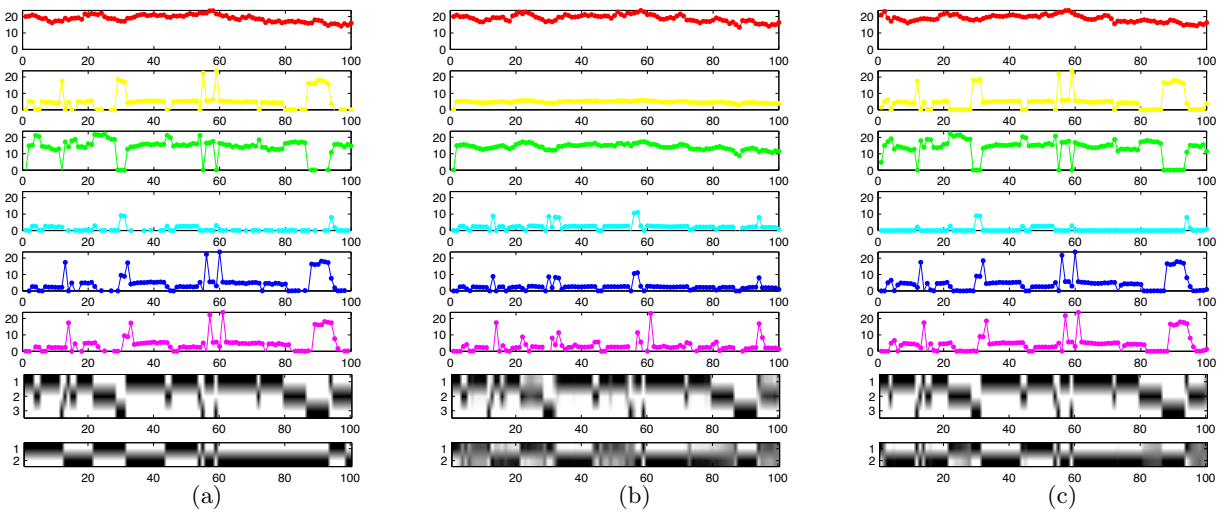

Figure 25.6: Given the observations from fig(25.5) we infer the flows and switch states of all the latent variables. **(a)**: The correct latent flows through time along with the switch variable state used to generate the data. The colours corresponds to the flows at the corresponding coloured edges/nodes in fig(25.4). **(b)**: Filtered flows based on a $I = 2$ Gaussian Sum forward pass approximation. Plotted are the 6 components of the vector $\langle \mathbf{h}_t | \mathbf{v}_{1:t} \rangle$ with the posterior distribution of the s_a and s_b traffic light states $p(s_t^a|\mathbf{v}_{1:t})$,$p(s_t^b|\mathbf{v}_{1:t})$ plotted below. **(c)**: Smoothed flows $\langle \mathbf{h}_t | \mathbf{v}_{1:T} \rangle$ and corresponding smoothed switch states $p(s_t|\mathbf{v}_{1:T})$ using a Gaussian Sum smoothing approximation (EC with $J = 1$).

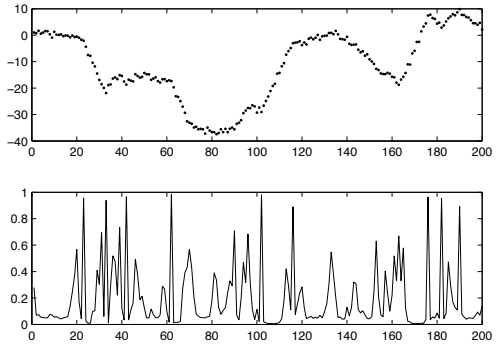

Figure 25.7: The top panel is a time series of 'prices'. The prices tend to keep going up or down with infrequent changes in the direction. Based on fitting a simple SLDS model to capture this kind of behaviour, the probability of a significant change in the price direction is given in the panel below based on the smoothed distribution $p(s_t = 2|v_{1:T})$.

25.5 Reset Models

Reset models are special switching models in which the switch state isolates the present from the past, resetting the position of the latent dynamics (these are also known as changepoint models). Whilst these models are rather general, it can be helpful to consider a specific model, and here we consider the SLDS changepoint model with two states. We use the state $s_t = 0$ to denote that the LDS continues with the standard dynamics. With $s_t = 1$, however, the continuous dynamics is reset to a prior:

$$p(\mathbf{h}_t|\mathbf{h}_{t-1}, s_t) = \begin{cases} p^0(\mathbf{h}_t|\mathbf{h}_{t-1}) & s_t = 0 \\ p^1(\mathbf{h}_t) & s_t = 1 \end{cases} \tag{25.5.1}$$

where

$$p^0(\mathbf{h}_t|\mathbf{h}_{t-1}) = \mathcal{N}\left(\mathbf{h}_t|\mathbf{A}\mathbf{h}_{t-1} + \boldsymbol{\mu}^0, \boldsymbol{\Sigma}^0\right), \qquad p^1(\mathbf{h}_t) = \mathcal{N}\left(\mathbf{h}_t|\boldsymbol{\mu}^1, \boldsymbol{\Sigma}^1\right) \tag{25.5.2}$$

similarly we write

$$p(\mathbf{v}_t|\mathbf{h}_t, s_t) = \begin{cases} p^0(\mathbf{v}_t|\mathbf{h}_t) & s_t = 0 \\ p^1(\mathbf{v}_t|\mathbf{h}_t) & s_t = 1 \end{cases} \tag{25.5.3}$$

For simplicity we assume the switch dynamics are first order Markov with transition $p(s_t|s_{t-1})$. Under this model the dynamics follows a standard LDS, but when $s_t = 1$, $\mathbf{h}_t$ is reset to a value drawn from a Gaussian distribution, independent of the past. Such models might be of interest in prediction where the time-series is

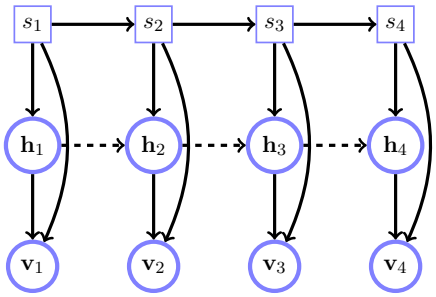

Figure 25.8: The independence structure of a reset model. Square nodes s_t denote the binary reset variables and $\mathbf{h}_t$ the continuous state. The $\mathbf{h}_t$ are continuous variables, and $\mathbf{v}_t$ continuous observations. If the dynamics resets, the dependence of the continuous $\mathbf{h}_t$ on the past is cut.

following a trend but suddenly changes and the past is forgotten. Whilst this may not seem like a big change to the model, this model is computationally more tractable, scaling with $O\left(T^2\right)$, compared to $O\left(T2^T\right)$ in the general two-state SLDS. To see this, consider the filtering recursion

$$\alpha(\mathbf{h}_t, s_t) \propto \int_{\mathbf{h}_{t-1}} \sum_{s_{t-1}} p(\mathbf{v}_t|\mathbf{h}_t, s_t)p(\mathbf{h}_t|\mathbf{h}_{t-1}, s_t)p(s_t|s_{t-1})\alpha(\mathbf{h}_{t-1}, s_{t-1}) \tag{25.5.4}$$

We now consider the two cases

$$\alpha(\mathbf{h}_t, s_t = 0) \propto \int_{\mathbf{h}_{t-1}} \sum_{s_{t-1}} p^0(\mathbf{v}_t|\mathbf{h}_t)p^0(\mathbf{h}_t|\mathbf{h}_{t-1})p(s_t = 0|s_{t-1})\alpha(\mathbf{h}_{t-1}, s_{t-1}) \tag{25.5.5}$$

$$\begin{aligned}\alpha(\mathbf{h}_t, s_t = 1) &\propto p^1(\mathbf{v}_t|\mathbf{h}_t)p^1(\mathbf{h}_t) \int_{\mathbf{h}_{t-1}} \sum_{s_{t-1}} p(s_t = 1|s_{t-1})\alpha(\mathbf{h}_{t-1}, s_{t-1}) \\ &\propto p^1(\mathbf{v}_t|\mathbf{h}_t)p^1(\mathbf{h}_t) \sum_{s_{t-1}} p(s_t = 1|s_{t-1})\alpha(s_{t-1})\end{aligned} \tag{25.5.6}$$

Equation(25.5.6) shows that $p(\mathbf{h}_t, s_t = 1|\mathbf{v}_{1:t})$ is not a mixture model in $\mathbf{h}_t$, but contains only a single component proportional to $p^1(\mathbf{v}_t|\mathbf{h}_t)p^1(\mathbf{h}_t)$. If we use this information in equation (25.5.5) we have

$$\begin{aligned}\alpha(\mathbf{h}_t, s_t = 0) \propto &\int_{\mathbf{h}_{t-1}} p^0(\mathbf{v}_t|\mathbf{h}_t)p^0(\mathbf{h}_t|\mathbf{h}_{t-1})p(s_t = 0|s_{t-1} = 0)\alpha(\mathbf{h}_{t-1}, s_{t-1} = 0) \\ &+ \int_{\mathbf{h}_{t-1}} p^0(\mathbf{v}_t|\mathbf{h}_t)p^0(\mathbf{h}_t|\mathbf{h}_{t-1})p(s_t = 0|s_{t-1} = 1)\alpha(\mathbf{h}_{t-1}, s_{t-1} = 1)\end{aligned} \tag{25.5.7}$$

Assuming $\alpha(\mathbf{h}_{t-1}, s_{t-1} = 0)$ is a mixture distribution with K components, then $\alpha(\mathbf{h}_t, s_t = 0)$ will be a mixture with $K + 1$ components. In general, therefore, $\alpha(\mathbf{h}_t, s_t = 0)$ will contain T components and $\alpha(\mathbf{h}_t, s_t = 1)$ a single component. As opposed to the full SLDS case, the number of components therefore grows only linearly with time, as opposed to exponentially. This means that the computational effort to perform exact filtering scales as $O\left(T^2\right)$. Smoothing can be achieved using the $\alpha - \beta$ approach, see exercise(25.3).

Run-length formalism

One may also describe reset models using a 'run-length' formalism using at each time t a latent variable r_t which describes the length of the current segment[3]. If there is a change, the run-length variable is reset to zero, otherwise it is increased by 1:

$$p(r_t|r_{t-1}) = \begin{cases} P_{cp} & r_t = 0 \\ 1 - P_{cp} & r_t = r_{t-1} + 1 \end{cases} \tag{25.5.8}$$

where P_{cp} is the probability of a reset (or 'changepoint'). The joint distribution is given by

$$p(v_{1:T}, r_{1:T}) = \prod_t p(r_t|r_{t-1})p(v_t|v_{1:t-1}, r_t), \qquad p(v_t|v_{1:t-1}, r_t) = p(v_t|v_{t-r_t:t-1}) \tag{25.5.9}$$

with the understanding that if $r_t = 0$ then $p(v_t|v_{t-r_t:t-1}) = p(v_t)$. The graphical model of this distribution is awkward to draw since the number of links depends on the run-length r_t. Predictions can be made using

$$p(v_{t+1}|v_{1:t}) = \sum_{r_t} p(v_{t+1}|v_{t-r_t:t})p(r_t|v_{1:t}) \tag{25.5.10}$$

where the filtered 'run-length' $p(r_t|v_{1:t})$ is given by the forward recursion:

$$\begin{aligned} p(r_t, v_{1:t}) &= \sum_{r_{t-1}} p(r_t, r_{t-1}, v_{1:t-1}, v_t) = \sum_{r_{t-1}} p(r_t, v_t|r_{t-1}, v_{1:t-1}) p(r_{t-1}, v_{1:t-1}) \\ &= \sum_{r_{t-1}} p(v_t|r_t, \cancel{r_{t-1}}, v_{1:t-1}) p(r_t|r_{t-1}, \cancel{v_{1:t-1}}) p(r_{t-1}, v_{1:t-1}) \\ &= \sum_{r_{t-1}} p(r_t|r_{t-1}) p(v_t|v_{t-r_t:t-1}) p(r_{t-1}, v_{1:t-1}) \end{aligned}$$

which shows that filtered inference scales with $O\left(T^2\right)$.

25.5.1 A Poisson reset model

The changepoint structure is not limited to conditionally Gaussian cases only. To illustrate this, we consider the following model[2]: At each time t, we observe a count y_t which we assume is Poisson distributed with an unknown positive intensity h. The intensity is constant, but at certain unknown times t, it jumps to a new value. The indicator variable c_t denotes whether time t is such a changepoint or not. Mathematically, the model is:

$$p(h_0) = \mathcal{G}(h_0; a_0, b_0) \tag{25.5.11}$$
$$p(c_t) = \mathcal{BE}(c_t; \pi) \tag{25.5.12}$$
$$p(h_t|h_{t-1}, c_t) = \mathbb{I}\left[c_t = 0\right] \delta(h_t, h_{t-1}) + \mathbb{I}\left[c_t = 1\right] \mathcal{G}(h_t; \nu, b) \tag{25.5.13}$$
$$p(v_t|h_t) = \mathcal{PO}(v_t; h_t) \tag{25.5.14}$$

The symbols $\mathcal{G}$, $\mathcal{BE}$ and $\mathcal{PO}$ denote the Gamma, Bernoulli and the Poisson distributions respectively:

$$\mathcal{G}(h; a, b) = \exp\left((a-1)\log h - bh - \log\Gamma(a) + a\log b\right) \tag{25.5.15}$$
$$\mathcal{BE}(c; \pi) = \exp\left(c\log\pi + (1-c)\log(1-\pi)\right) \tag{25.5.16}$$
$$\mathcal{PO}(v; h) = \exp\left(v\log h - h - \log\Gamma(v+1)\right) \tag{25.5.17}$$

Given observed counts $v_{1:T}$, the task is to find the posterior probability of a change and the associated intensity levels for each region between two consecutive changepoints. Plugging the above definitions in the generic updates equation (25.5.5) and equation (25.5.6), we see that $\alpha(h_t, c_t = 0)$ is a Gamma potential, and that $\alpha(g_t, c_t = 1)$ is a mixture of Gamma potentials, where a Gamma potential is defined as

$$\phi(h) = e^l \mathcal{G}(h; a, b) \tag{25.5.18}$$

via the triple (a, b, l). For the corrector update step we need to calculate the product of a Poisson term with the observation model $p(v_t|h_t) = \mathcal{PO}(v_t; h_t)$. A useful property of the Poisson distribution is that, given the observation, the latent variable is Gamma distributed:

$$\mathcal{PO}(v; h) = v\log h - h - \log\Gamma(v+1) \tag{25.5.19}$$
$$= (v+1-1)\log h - h - \log\Gamma(v+1) \tag{25.5.20}$$
$$= \mathcal{G}(h; v+1, 1) \tag{25.5.21}$$

Hence, the update equation requires multiplication of two Gamma potentials. A nice property of the Gamma density is that the product of two Gamma densities is also a Gamma potential:

$$(a_1, b_1, l_1) \times (a_2, b_2, l_2) = (a_1 + a_2 - 1, b_1 + b_2, l_1 + l_2 + g(a_1, b_1, a_2, b_2)) \tag{25.5.22}$$

where

$$g(a_1, b_1, a_2, b_2) \equiv \log\frac{\Gamma(a_1 + a_2 - 1)}{\Gamma(a_1)\Gamma(a_2)} + \log(b_1 + b_2) + a_1\log(b_1/(b_1 + b_2)) + a_2\log(b_2/(b_1 + b_2)) \tag{25.5.23}$$

The α recursions for this reset model are therefore closed in the space of a mixture of Gamma potentials, with an additional Gamma potential in the mixture at each timestep. A similar approach can be used to form the smoothing recursions, see exercise(25.3).

[2]This example is due to Taylan Cemgil.

Example 25.3 (Coal mining disasters)**.** We illustrate the algorithm on the coal mining disaster dataset [147]. The data set consists of the number of deadly coal-mining disasters in England per year over a time span of 112 years from 1851 to 1962. It is widely agreed in the statistical literature that a change in the intensity (the expected value of the number of disasters) occurs around the year 1890, after new health and safety regulations were introduced. In fig(25.9) we show the marginals $p(h_t|y_{1:T})$ along with the filtering density. Note that we are not constraining the number of changepoints and in principle allow any number. The smoothed density suggests a sharp decrease around $t = 1890$.

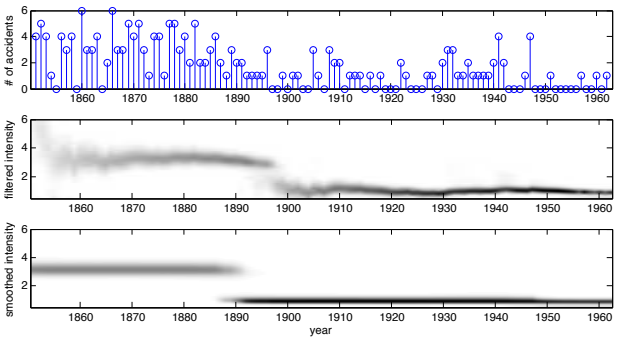

Figure 25.9: Estimation of change points. (Top) coal mining disaster dataset. (Middle) Filtered estimate of the marginal intensity $p(h_t|v_{1:t})$ and (Bottom) smoothed estimate $p(h_t|v_{1:T})$. We evaluate these mixture of Gamma distributions on a fixed grid of h and show the density as a function of t. Here, darker color means higher probability.

25.5.2 Reset-HMM-LDS

The reset model defined by equations (25.5.1,25.5.3) above is useful in many applications, but is limited since only a single dynamical model is considered. An important extension is to consider a set of available dynamical models, indexed by $s_t \in \{1, \ldots, S\}$, with a reset that cuts dependency of the continuous variable on the past[95, 57]:

$$p(\mathbf{h}_t|\mathbf{h}_{t-1}, s_t, c_t) = \begin{cases} p^0(\mathbf{h}_t|\mathbf{h}_{t-1}, s_t) & c_t = 0 \\ p^1(\mathbf{h}_t|s_t) & c_t = 1 \end{cases} \tag{25.5.24}$$

The states s_t follow a Markovian dynamics $p(s_t|s_{t-1}, c_{t-1})$, see fig(25.10). A reset occurs if the state s_t changes, otherwise, no reset occurs:

$$p(c_t = 1|s_t, s_{t-1}) = \mathbb{I}[s_t \neq s_{t-1}] \tag{25.5.25}$$

The computational complexity of filtering for this model is $O\left(S^2T^2\right)$ which can be understood by analogy with the reset α recursions, equations(25.5.5,25.5.6) on replacing h_t by (h_t, s_t). To see this we consider the filtering recursion for the two cases

$$\alpha(\mathbf{h}_t, s_t, c_t = 0) = \int_{\mathbf{h}_{t-1}} \sum_{s_{t-1}, c_{t-1}} p^0(\mathbf{v}_t|\mathbf{h}_t, s_t) p^0(\mathbf{h}_t|\mathbf{h}_{t-1}, s_t) p(s_t|s_{t-1}, c_{t-1}) p(c_t = 0|s_t, s_{t-1}) \alpha(\mathbf{h}_{t-1}, c_{t-1}) \tag{25.5.26}$$

$$\begin{aligned} \alpha(\mathbf{h}_t, s_t, c_t = 1) &= \int_{\mathbf{h}_{t-1}} \sum_{s_{t-1}, c_{t-1}} p^1(\mathbf{v}_t|\mathbf{h}_t, s_t) p^1(\mathbf{h}_t|s_t) p(s_t|s_{t-1}, c_t) p(c_t = 1|s_t, s_{t-1}) \alpha(\mathbf{h}_{t-1}, s_{t-1}, c_{t-1}) \\ &= p^1(\mathbf{v}_t|\mathbf{h}_t, s_t) p^1(\mathbf{h}_t|s_t) \sum_{s_{t-1}, c_{t-1}} p(c_t = 1|s_t, s_{t-1}) p(s_t|s_{t-1}, c_{t-1}) \alpha(s_{t-1}, c_{t-1}) \end{aligned} \tag{25.5.27}$$

From equation (25.5.27) we see that $\alpha(\mathbf{h}_t, s_t, c_t = 1)$ contains only a single component proportional to $p^1(\mathbf{v}_t|\mathbf{h}_t, s_t)p^1(\mathbf{h}_t|s_t)$. This is therefore exactly analogous to the standard reset model, except that we need now to index a set of messages with s_t, therefore each message taking $O(S)$ steps to compute. The computational effort to perform exact filtering scales as $O\left(S^2T^2\right)$.

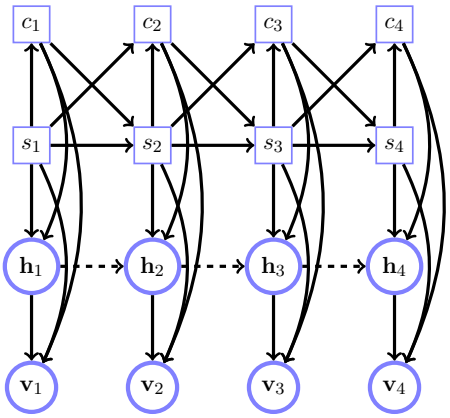

Figure 25.10: The independence structure of a reset-HMM-LDS model. Square nodes $c_t \in \{0, 1\}$ denote reset variables; $\mathbf{h}_t$ are continuous latent variables, and $\mathbf{v}_t$ continuous observations. The discrete state $s_t \in \{1, \ldots, S\}$ determines which Linear Dynamical system from a finite set of Linear Dynamical systems is operational at time t.

25.6 Code

`SLDSforward.m`: SLDS forward
`SLDSbackward.m`: SLDS backward (Expectation Correction)
`mix2mix.m`: Collapse a mixture of Gaussians to a smaller mixture of Gaussians
`SLDSmargGauss.m`: Marginalise an SLDS Gaussian mixture
`logeps.m`: Logarithm with offset to deal with log(0)
`demoSLDStraffic.m`: Demo of Traffic Flow using a switching Linear Dynamical System

25.7 Summary

- The switching linear dynamical system is a marriage of a discrete state HMM with the continuous latent state linear dynamical system. It is able to model discrete jumps in an underlying continuous process, finding application in a large variety of domains from finance to speech processing.
- The classical inference problems in the SLDS are formally intractable since representing the messages requires an exponential amount of space.
- Many approximation methods have been developed for the SLDS and in this chapter we described a robust deterministic method based on a mixture of Gaussians representation.
- Reset models are such that the continuous variable forgets the past when reset. Unlike the SLDS such models are much more amenable to exact inference. Extensions include the reset-HMM which allows for a set of discrete and continuous states in which a special discrete state resets the continuous states.

25.8 Exercises

Exercise 25.1. *Consider the setup described in example(25.2), for which the full SLDS model is given in* `SLDSpricemodel.m`*, following the notation used in* `demoSLDStraffic.m`*. Given the data in the vector* `v` *your task is to fit a prediction model to the data. To do so, approximate the filtered distribution* $p(h(t), s(t)|v_{1:t})$ *using a mixture of* $I = 2$ *components. The prediction of the price at the next day is then*

$$v^{pred}(t+1) = \langle h_1(t) + h_2(t) \rangle_{p(h(t)|v_{1:t})} \tag{25.8.1}$$

where $p(h(t)|v_{1:t}) = \sum_{s_t} p(h(t), s_t|v_{1:t})$.

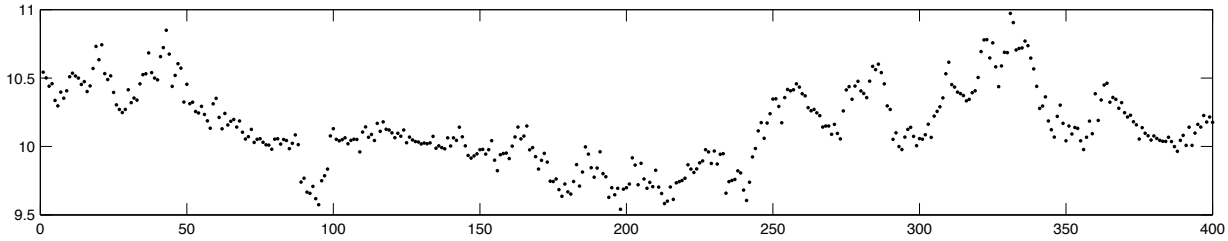

Figure 25.11: Data from an intermittent mean-reverting process. See exercise(25.2).

1. *Compute the mean prediction error*
   `mean_abs_pred_error=mean(abs(vpred(2:end)-v(2:end)))`
2. *Compute the mean naive prediction error*
   `mean_abs_pred_error_naive=mean(abs(v(1:end-1)-v(2:end)))`
   which corresponds to saying that tomorrow's price will be the same as today's.

You might find `SLDSmargGauss.m` *of interest.*

Exercise 25.2. *The data in fig(25.11) are observed prices from an intermittent mean-reverting process, contained in* `meanrev.mat`*. There are two states $S = 2$. There is a true (latent) price p_t and an observed price v_t (which is plotted). When $s = 1$, the true underlying price reverts back towards the mean $m = 10$ with rate $r = 0.9$. Otherwise the true price follows a random walk:*

$$p_t = \begin{cases} r(p_{t-1} - m) + m + \eta_t^p & s_t = 1 \\ p_{t-1} + \eta_t^p & s_t = 2 \end{cases} \tag{25.8.2}$$

where

$$\eta_t^p \sim \begin{cases} \mathcal{N}(\eta_t^p | 0, 0.0001) & s_t = 1 \\ \mathcal{N}(\eta_t^p | 0, 0.01) & s_t = 2 \end{cases} \tag{25.8.3}$$

The observed price v_t is related to the unknown price p_t by

$$v_t \sim \mathcal{N}(v_t | p_t, 0.001) \tag{25.8.4}$$

It is known that 95% of the time s_{t+1} is in the same state as at time $t > 1$ and that at time $t = 1$ either state of s is equally likely. Also at $t = 1$, $p_1 \sim \mathcal{N}(p_1 | m, 0.1)$. Based on this information, and using Gaussian Sum filtering with $I = 2$ components (use `SLDSforward.m`*), what is the probability at time $t = 280$ that the dynamics is following a random walk, $p(s_{280} = 2 | v_{1:280})$? Repeat this computation for smoothing $p(s_{280} = 2 | v_{1:400})$ based on using Expectation Correction with $I = J = 2$ components.*

Exercise 25.3. *We here derive a smoothing recursion based on the $\alpha - \beta$ approach for the reset-LDS described in section(25.5). Smoothing can be achieved using*

$$p(\mathbf{h}_t, s_t | \mathbf{v}_{1:T}) \propto \underbrace{p(\mathbf{h}_t, s_t | \mathbf{v}_{1:t})}_{\alpha(\mathbf{h}_t, s_t)} \underbrace{p(\mathbf{v}_{t+1:T} | \mathbf{h}_t, s_t)}_{\beta(\mathbf{h}_t, s_t)} \tag{25.8.5}$$

From the formal β recursion, we have

$$\beta(\mathbf{h}_{t-1}, s_{t-1}) = \sum_{s_t} \int_{\mathbf{h}_t} p(\mathbf{v}_t | \mathbf{h}_t, s_t) p(\mathbf{h}_t | \mathbf{h}_{t-1}, s_t) p(s_t | s_{t-1}) \beta(\mathbf{h}_t, s_t) \tag{25.8.6}$$

Show that the rhs can be written as

$$\underbrace{p(s_t = 0 | s_{t-1}) \int_{\mathbf{h}_t} p^0(\mathbf{v}_t | \mathbf{h}_t) p^0(\mathbf{h}_t | \mathbf{h}_{t-1}) \beta(\mathbf{h}_t, s_t = 0)}_{\beta^0(\mathbf{h}_{t-1}, s_{t-1})} + \underbrace{p(s_t = 1 | s_{t-1}) \int_{\mathbf{h}_t} p^1(\mathbf{v}_t | \mathbf{h}_t) p^1(\mathbf{h}_t) \beta(\mathbf{h}_t, s_t = 1)}_{\beta^1(s_{t-1})} \tag{25.8.7}$$

Writing

$$\beta(\mathbf{h}_t, s_t) = \beta^0(\mathbf{h}_t, s_t) + \beta^1(s_t) \tag{25.8.8}$$

derive then the recursions

$$\beta^0(\mathbf{h}_{t-1}, s_{t-1}) = p(s_t = 0|s_{t-1}) \int_{\mathbf{h}_t} p^0(\mathbf{v}_t|\mathbf{h}_t) p^0(\mathbf{h}_t|\mathbf{h}_{t-1}) \left[\beta^0(\mathbf{h}_t, s_t = 0) + \beta^1(s_t = 0)\right] \tag{25.8.9}$$

and

$$\beta^1(s_{t-1}) = p(s_t = 1|s_{t-1}) \int_{\mathbf{h}_t} p^1(\mathbf{v}_t|\mathbf{h}_t) p^1(\mathbf{h}_t) \left[\beta^0(\mathbf{h}_t, s_t = 1) + \beta^1(s_t = 1)\right] \tag{25.8.10}$$

The β^1 contribution is simply a scalar, so that its complexity of representation is fixed. According to the above β^0 recursion, we include an additional component in the representation of β^0 with each timestep that we go backwards in time. Thus the number of components to represent $\beta^0(\mathbf{h}_t, s_t)$ will be $O(T-t)$, since at time T, we may define $\beta(h_T, s_t) = 1$. This means that the term $p(\mathbf{h}_t, s_t, \mathbf{v}_{1:T}) = \alpha(\mathbf{h}_t, s_t)\beta(\mathbf{h}_t, s_t)$ will contain $O(t(T-t))$ components. To form a complete smoothing pass over all time therefore takes $O\left(T^3\right)$ time.