Chapter 5

Variational Bayesian Linear Dynamical Systems

5.1 Introduction

This chapter is concerned with the variational Bayesian treatment of Linear Dynamical Systems (LDSs), also known as linear-Gaussian state-space models (SSMs). These models are widely used in the fields of signal filtering, prediction and control, because: (1) many systems of interest can be approximated using linear systems, (2) linear systems are much easier to analyse than nonlinear systems, and (3) linear systems can be estimated from data efficiently. State-space models assume that the observed time series data was generated from an underlying sequence of unobserved (hidden) variables that evolve with Markovian dynamics across successive time steps. The filtering task attempts to infer the likely values of the hidden variables that generated the current observation, given a sequence of observations up to and including the current observation; the prediction task tries to simulate the unobserved dynamics one or many steps into the future to predict a future observation.

The task of deciding upon a suitable dimension for the hidden state space remains a difficult problem. Traditional methods, such as early stopping, attempt to reduce generalisation error by terminating the learning algorithm when the error as measured on a hold-out set begins to increase. However the hold-out set error is a noisy quantity and for a reliable measure a large set of data is needed. We would prefer to learn from all the available data, in order to make predictions. We also want to be able to obtain posterior distributions over all the parameters in the model in order to quantify our uncertainty.

We have already shown in chapter 4 that we can infer the dimensionality of the hidden variable space (i.e. the number of factors) in a mixture of factor analysers model, by placing priors on

the factor loadings which then implement automatic relevance determination. Linear-Gaussian state-space models can be thought of as factor analysis through time with the hidden factors evolving with noisy linear dynamics. A variational Bayesian treatment of these models provides a novel way to learn their structure, i.e. to identify the optimal dimensionality of their state space.

With suitable priors the LDS model is in the conjugate-exponential family. This chapter presents an example of variational Bayes applied to a conjugate-exponential model, which therefore results in a VBEM algorithm which has an approximate inference procedure with the same complexity as the MAP/ML counterpart, as explained in chapter 2. Unfortunately, the implementation is not as straightforward as in other models, for example the Hidden Markov Model of chapter 3, as some subparts of the parameter-to-natural parameter mapping are non-invertible.

The rest of this chapter is written as follows. In section 5.2 we review the LDS model for both the standard and input-dependent cases, and specify conjugate priors over all the parameters. In 5.3 we use the VB lower bounding procedure to approximate the Bayesian integral for the marginal likelihood of a sequence of data under a particular model, and derive the VBEM algorithm. The VBM step is straightforward, but the VBE step is much more interesting and we fully derive the forward and backward passes analogous to the Kalman filter and Rauch-Tung-Striebel smoothing algorithms, which we call the *variational Kalman filter* and *smoother* respectively. In this section we also discuss hyperparameter learning (including optimisation of automatic relevance determination hyperparameters), and also show how the VB lower bound can be computed. In section 5.4 we demonstrate the model's ability to discover meaningful structure from synthetically generated data sets (in terms of the dimension of the hidden state space etc.). In section 5.5 we present a very preliminary application of the VB LDS model to real DNA microarray data, and attempt to discover underlying mechanisms in the immune response of human T-lymphocytes, starting from T-cell receptor activation through to gene transcription events in the nucleus. In section 5.6 we suggest extensions to the model and possible future work, and in section 5.7 we provide some conclusions.

5.2 The Linear Dynamical System model

5.2.1 Variables and topology

In state-space models (SSMs), a sequence $(\mathbf{y}_1, \ldots, \mathbf{y}_T)$ of p-dimensional real-valued observation vectors, denoted $\mathbf{y}_{1:T}$, is modelled by assuming that at each time step t, $\mathbf{y}_t$ was generated from a k-dimensional real-valued hidden state variable $\mathbf{x}_t$, and that the sequence of $\mathbf{x}$'s follow

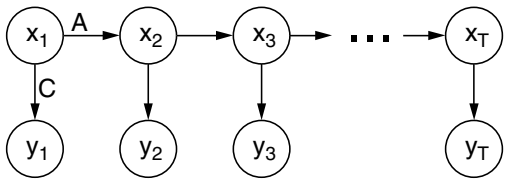


Figure 5.1: Graphical model representation of a state-space model. The hidden variables $\mathbf{x}_t$ evolve with Markov dynamics according to parameters in A, and at each time step generate an observation $\mathbf{y}_t$ according to parameters in C.

a first-order Markov process. The joint probability of a sequence of states and observations is therefore given by:

$$p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) = p(\mathbf{x}_1)p(\mathbf{y}_1 \,|\, \mathbf{x}_1) \prod_{t=2}^{T} p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})p(\mathbf{y}_t \,|\, \mathbf{x}_t) \;. \tag{5.1}$$

This factorisation of the joint probability can be represented by the graphical model shown in figure 5.1. For the moment we consider just a single sequence, not a batch of i.i.d. sequences. For ML and MAP learning there is a straightforward extension for learning multiple sequences; for VB learning the extensions are outlined in section 5.3.8.

The form of the distribution $p(\mathbf{x}_1)$ over the first hidden state is Gaussian, and is described and explained in more detail in section 5.2.2. We focus on models where both the dynamics, $p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})$, and output functions, $p(\mathbf{y}_t \,|\, \mathbf{x}_t)$, are linear and time-invariant and the distributions of the state evolution and observation noise variables are Gaussian, i.e. linear-Gaussian state-space models:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + \mathbf{w}_t \;, \qquad \mathbf{w}_t \sim \mathrm{N}(\mathbf{0}, Q) \tag{5.2}$$

$$\mathbf{y}_t = C\mathbf{x}_t + \mathbf{v}_t \;, \qquad \mathbf{v}_t \sim \mathrm{N}(\mathbf{0}, R) \tag{5.3}$$

where A $(k \times k)$ is the state dynamics matrix, C $(p \times k)$ is the observation matrix, and Q $(k \times k)$ and R $(p \times p)$ are the covariance matrices for the state and output noise variables $\mathbf{w}_t$ and $\mathbf{v}_t$. The parameters A and C are analogous to the transition and emission matrices respectively in a Hidden Markov Model (see chapter 3). Linear-Gaussian state-space models can be thought of as factor analysis where the low-dimensional (latent) factor vector at one time step diffuses linearly with Gaussian noise to the next time step.

We will use the terms 'linear dynamical system' (LDS) and 'state-space model' (SSM) interchangeably throughout this chapter, although they emphasise different properties of the model. LDS emphasises that the dynamics are linear – such models can be represented either in state-space form or in input-output form. SSM emphasises that the model is represented as a latent-variable model (i.e. the observables are generated via some hidden states). SSMs can be non-

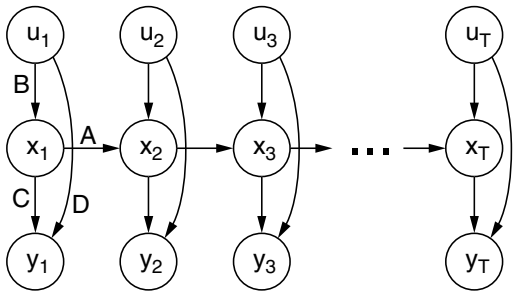


Figure 5.2: The graphical model for linear dynamical systems with inputs.

linear in general; here it should be assumed that we refer to linear models with Gaussian noise except if stated otherwise.

A straightforward extension to this model is to allow both the dynamics and observation model to include a dependence on a series of d-dimensional driving inputs $\mathbf{u}_{1:T}$:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{u}_t + \mathbf{w}_t \tag{5.4}$$

$$\mathbf{y}_t = C\mathbf{x}_t + D\mathbf{u}_t + \mathbf{v}_t \ . \tag{5.5}$$

Here B $(k \times d)$ and D $(p \times d)$ are the input-to-state and input-to-observation matrices respectively. If we now augment the driving inputs with a constant bias, then this input driven model is able to incorporate an arbitrary origin displacement for the hidden state dynamics, and also can induce a displacement in the observation space. These displacements can be learnt as parameters of the input-to-state and input-to-observation matrices.

Figure 5.2 shows the graphical model for an input-dependent linear dynamical system. An input-dependent model can be used to model control systems. Another possible way in which the inputs can be utilised is to feedback the outputs (data) from previous time steps in the sequence into the inputs for the current time step. This means that the hidden state can concentrate on modelling hidden factors, whilst the Markovian dependencies between successive *outputs* are modelled using the output-input feedback construction. We will see a good example of this type of application in section 5.5, where we use it to model gene expression data in a DNA microarray experiment.

On a point of notational convenience, the probability statements in the later derivations leave implicit the dependence of the dynamics and output processes on the driving inputs, since for each sequence they are fixed and merely modulate the processes at each time step. Their omission keeps the equations from becoming unnecessarily complicated.

Without loss of generality we can set the hidden state evolution noise covariance, Q, to the identity matrix. This is possible since an arbitrary noise covariance can be incorporated into the state dynamics matrix A, and the hidden state rescaled and rotated to be made commensurate with

this change (see Roweis and Ghahramani, 1999, page 2 footnote); these changes are possible since the hidden state is unobserved, by definition. This is the case in the maximum likelihood scenario, but in the MAP or Bayesian scenarios this degeneracy is lost since various scalings in the parameters will be differently penalised under the parameter priors (see section 5.2.2 below).

The remaining parameter of a linear-Gaussian state-space model is the covariance matrix, R, of the Gaussian output noise, $\mathbf{v}_t$. In analogy with factor analysis we assume this to be diagonal. Unlike the hidden state noise, Q, there is no degeneracy in R since the data is observed, and therefore its scaling is fixed and needs to be learnt.

For notational convenience we collect the above parameters into a single parameter vector for the model: $\boldsymbol{\theta} = (A, B, C, D, R)$.

We now turn to considering the LDS model for a Bayesian analysis. From (5.1), the complete-data likelihood for linear-Gaussian state-space models is Gaussian, which is in the class of exponential family distributions, thus satisfying condition 1 (2.80). In order to derive a variational Bayesian algorithm by applying the results in chapter 2 we now build on the model by defining conjugate priors over the parameters according to condition 2 (2.88).

5.2.2 Specification of parameter and hidden state priors

The description of the priors in this section may be made more clear by referring to figure 5.3. The forms of the following prior distributions are motivated by conjugacy (condition 2, (2.88)). By writing every term in the complete-data likelihood (5.1) explicitly, we notice that the likelihood for state-space models factors into a product of terms for every *row* of each of the dynamics-related and output-related matrices, and the priors can therefore be factorised over the hidden variable and observed data dimensions.

The prior over the output noise covariance matrix R, which is assumed diagonal, is defined through the precision vector $\boldsymbol{\rho}$ such that $R^{-1} = \mathrm{diag}\,(\boldsymbol{\rho})$. For conjugacy, each dimension of $\boldsymbol{\rho}$ is assumed to be gamma distributed with hyperparameters a and b:

$$p(\boldsymbol{\rho} \,|\, a, b) = \prod_{s=1}^{p} \frac{b^a}{\Gamma(a)} \rho_s^{a-1} \exp\{-b\rho_s\}. \tag{5.6}$$

More generally, we could let R be a full covariance matrix and still be conjugate: its inverse $V = R^{-1}$ would be given a Wishart distribution with parameter S and degrees of freedom ν:

$$p(V \,|\, \nu, S) \propto |V|^{(\nu-p-1)/2} \exp\left[-\frac{1}{2}\mathrm{tr}\; VS^{-1}\right], \tag{5.7}$$

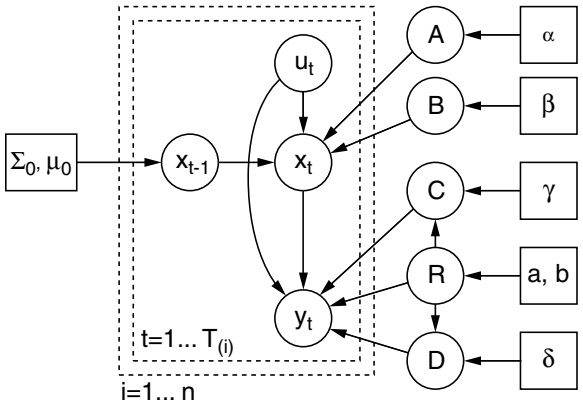


Figure 5.3: Graphical model representation of a Bayesian state-space model. Each sequence $\{\mathbf{y}_1, \ldots, \mathbf{y}_{T_i}\}$ is now represented succinctly as the (inner) plate over T_i pairs of hidden variables, each presenting the cross-time dynamics and output process. The second (outer) plate is over the data set of size n sequences. For the most part of the derivations in this chapter we restrict ourselves to $n = 1$, and $T_n = T$. Note that the plate notation used here is non-standard since both $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ have to be included in the plate to denote the dynamics.

where tr is the matrix trace operator. This more general form is not adopted in this chapter as we wish to maintain a parallel between the output model for state-space models and the factor analysis model (as described in chapter 4).

Priors on A, B, C and D

The row vector $\mathbf{a}_{(j)}^\top$ is used to denote the jth row of the dynamics matrix, A, and is given a zero mean Gaussian prior with precision equal to $\mathrm{diag}\,(\boldsymbol{\alpha})$, which corresponds to axis-aligned covariance and can possibly be non-spherical. Each row of C, denoted $\mathbf{c}_{(s)}^\top$, is given a zero-mean Gaussian prior with precision matrix equal to $\mathrm{diag}\,(\rho_s \boldsymbol{\gamma})$. The dependence of the precision of $\mathbf{c}_{(s)}$ on the noise output precision ρ_s is motivated by conjugacy (as can be seen from the explicit complete-data likelihood), and intuitively this prior links the scale of the signal to the noise. We place similar priors on the rows of the input-related matrices B and D, introducing two more hyperparameter vectors $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$. A useful notation to summarise these forms is

$$p(\mathbf{a}_{(j)} \,|\, \boldsymbol{\alpha}) = \mathrm{N}(\mathbf{a}_{(j)} \,|\, \mathbf{0}, \mathrm{diag}\,(\boldsymbol{\alpha})^{-1}) \tag{5.8}$$

$$p(\mathbf{b}_{(j)} \,|\, \boldsymbol{\beta}) = \mathrm{N}(\mathbf{b}_{(j)} \,|\, \mathbf{0}, \mathrm{diag}\,(\boldsymbol{\beta})^{-1}) \qquad \text{for } j = 1, \ldots, k \tag{5.9}$$

$$p(\mathbf{c}_{(s)} \,|\, \rho_s, \boldsymbol{\gamma}) = \mathrm{N}(\mathbf{c}_{(s)} \,|\, \mathbf{0}, \rho_s^{-1}\, \mathrm{diag}\,(\boldsymbol{\gamma})^{-1}) \tag{5.10}$$

$$p(\mathbf{d}_{(s)} \,|\, \rho_s, \boldsymbol{\delta}) = \mathrm{N}(\mathbf{d}_{(s)} \,|\, \mathbf{0}, \rho_s^{-1}\, \mathrm{diag}\,(\boldsymbol{\delta})^{-1}) \tag{5.11}$$

$$p(\rho_s \,|\, a, b) = \mathrm{Ga}(\rho_s \,|\, a, b) \qquad \text{for } s = 1, \ldots, p \tag{5.12}$$

such that $\mathbf{a}_{(j)}$ etc. are column vectors.

The Gaussian priors on the transition (A) and output (C) matrices can be used to perform 'automatic relevance determination' (ARD) on the hidden dimensions. As an example consider the matrix C which contains the linear embedding factor loadings for each factor in each of its columns: these factor loadings induce a high dimensional oriented covariance structure in the data ($CC^\top$), based on an embedding of low-dimensional axis-aligned (unit) covariance. Let us first fix the hyperparameters $\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_k\}$. As the parameters of the C matrix are learnt, the prior will favour entries close to zero since its mean is zero, and the degree with which the prior enforces this zero-preference varies across the columns depending on the size of the precisions in $\boldsymbol{\gamma}$. As learning continues, the burden of modelling the covariance in the p output dimensions will be gradually shifted onto those hidden dimensions for which the entries in $\boldsymbol{\gamma}$ are smallest, thus resulting in the least penalty under the prior for non-zero factor loadings. When the hyperparameters are updated to reflect this change, the unequal sharing of the output covariance is further exacerbated. The limiting effect as learning progresses is that some columns of C become zero, coinciding with the respective hyperparameters tending to infinity. This implies that those hidden state dimensions do not contribute to the covariance structure of data, and so can be removed entirely from the output process.

Analogous ARD processes can be carried out for the dynamics matrix A. In this case, if the jth column of A should become zero, this implies that the jth hidden dimension at time $t-1$ is not involved in generating the hidden state at time t (the rank of the transformation A is reduced by 1). However the jth hidden dimension may still be of use in producing covariance structure in the data via the modulatory input at each time step, and should not necessarily be removed unless the entries of the C matrix also suggest this.

For the input-related parameters in B and D, the ARD processes correspond to selecting those particular inputs that are relevant to driving the dynamics of the hidden state (through $\boldsymbol{\beta}$), and selecting those inputs that are needed to directly modulate the observed data (through $\boldsymbol{\delta}$). For example the (constant) input bias that we use here to model an offset in the data mean will almost certainly always remain non-zero, with a correspondingly small value in $\boldsymbol{\delta}$, unless the mean of the data is insignificantly far from zero.

Traditionally, the prior over the hidden state sequence is expressed as a Gaussian distribution directly over the first hidden state $\mathbf{x}_1$ (see, for example Ghahramani and Hinton, 1996a, equation (6)). For reasons that will become clear when later analysing the equations for learning the parameters of the model, we choose here to express the prior over the first hidden state indirectly through a prior over an auxiliary hidden state at time $t = 0$, denoted $\mathbf{x}_0$, which is Gaussian distributed with mean $\boldsymbol{\mu}_0$ and covariance Σ_0:

$$p(\mathbf{x}_0 \,|\, \boldsymbol{\mu}_0, \Sigma_0) = \mathrm{N}(\mathbf{x}_0 \,|\, \boldsymbol{\mu}_0, \Sigma_0) \,. \tag{5.13}$$

This induces a prior over $\mathbf{x}_1$ via the the state dynamics process:

$$p(\mathbf{x}_1 \mid \boldsymbol{\mu}_0, \Sigma_0, \boldsymbol{\theta}) = \int d\mathbf{x}_0 \; p(\mathbf{x}_0 \mid \boldsymbol{\mu}_0, \Sigma_0) p(\mathbf{x}_1 \mid \mathbf{x}_0, \boldsymbol{\theta}) \tag{5.14}$$

$$= \mathrm{N}(\mathbf{x}_1 \mid A\boldsymbol{\mu}_0 + B\mathbf{u}_1, A^\top \Sigma_0 A + Q) \; . \tag{5.15}$$

Although not constrained to be so, in this chapter we work with a prior covariance Σ_0 that is a multiple of the identity.

The marginal likelihood can then be written

$$p(\mathbf{y}_{1:T}) = \int dA \, dB \, dC \, dD \, d\boldsymbol{\rho} \, d\mathbf{x}_{0:T} \; p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}, \mathbf{y}_{1:T}) \; . \tag{5.16}$$

All hyperparameters can be optimised during learning (see section 5.3.6). In section 5.4 we present results of some experiments in which we show the variational Bayesian approach successfully determines the structure of state-space models learnt from synthetic data, and in section 5.5 we present some very preliminary experiments in which we attempt to use hyperparameter optimisation mechanisms to elucidate underlying interactions amongst genes in DNA microarray time-series data.

A fully hierarchical Bayesian structure

Depending on the task at hand we should consider how full a Bayesian analysis we require. As the model specification stands, there is the problem that the number of free parameters to be 'fit' increases with the complexity of the model. For example, if the number of hidden dimensions were increased then, even though the parameters of the dynamics (A), output (C), input-to-state (B), and input-to-observation (D) matrices are integrated out, the size of the $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$, $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ hyperparameters have increased, providing more parameters to fit. Clearly, the more parameters that are fit the more one departs from the Bayesian inference framework and the more one risks overfitting. But, as pointed out in MacKay (1995), these extra hyperparameters themselves cannot overfit the noise in the data, since it is only the parameters that can do so.

If the task at hand is structure discovery, then the presence of extra hyperparameters should not affect the returned structure. However if the task is model comparison, that is comparing the marginal likelihoods for models with different numbers of hidden state dimensions for example, or comparing differently structured Bayesian models, then optimising over more hyperparameters will introduce a bias favouring more complex models, unless they themselves are integrated out.

The proper marginal likelihood to use in this latter case is that which further integrates over the hyperparameters with respect to some hyperprior which expresses our subjective beliefs over

the distribution of these hyperparameters. This is necessary for the ARD hyperparameters, and also for the hyperparameters governing the prior over the hidden state sequence, $\boldsymbol{\mu}_0$ and Σ_0, whose number of free parameters are functions of the dimensionality of the hidden state, k. For example, the ARD hyperparameter for each matrix A, B, C, D would be given a separate spherical gamma hyperprior, which is conjugate:

$$\boldsymbol{\alpha} \sim \prod_{j=1}^{k} \mathrm{Ga}(\alpha_j \,|\, a_\alpha, b_\alpha) \tag{5.17}$$

$$\boldsymbol{\beta} \sim \prod_{c=1}^{d} \mathrm{Ga}(\beta_c \,|\, a_\beta, b_\beta) \tag{5.18}$$

$$\boldsymbol{\gamma} \sim \prod_{j=1}^{k} \mathrm{Ga}(\gamma_j \,|\, a_\gamma, b_\gamma) \tag{5.19}$$

$$\boldsymbol{\delta} \sim \prod_{c=1}^{d} \mathrm{Ga}(\delta_c \,|\, a_\delta, b_\delta)\ . \tag{5.20}$$

The hidden state hyperparameters would be given spherical Gaussian and spherical inverse-gamma hyperpriors:

$$\boldsymbol{\mu}_0 \sim \mathrm{N}(\boldsymbol{\mu}_0 \,|\, \mathbf{0}, b_{\boldsymbol{\mu}_0}\mathrm{I}) \tag{5.21}$$

$$\Sigma_0 \sim \prod_{j=1}^{k} \mathrm{Ga}(\Sigma_{0\,jj}^{-1} \,|\, a_{\Sigma_0}, b_{\Sigma_0})\ . \tag{5.22}$$

Inverse-Wishart hyperpriors for Σ_0 are also possible. For the most part of this chapter we omit this fuller hierarchy to keep the exposition clearer, and only perform experiments aimed at structure discovery using ARD as opposed to model comparison between this and other Bayesian models. Towards the end of the chapter there is a brief note on how the fuller Bayesian hierarchy affects the algorithms for learning.

Origin of the intractability with Bayesian learning

Since A, B, C, D, $\boldsymbol{\rho}$ and $\mathbf{x}_{0:T}$ are all unknown, given a sequence of observations $\mathbf{y}_{1:T}$, an exact Bayesian treatment of SSMs would require computing marginals of the posterior over parameters and hidden variables, $p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T} \,|\, \mathbf{y}_{1:T})$. This posterior contains interaction terms up to *fifth order*; we can see this by considering the terms in (5.1) for the case of LDS models which, for example, contain terms in the exponent of the form $-\frac{1}{2}\mathbf{x}_t^\top C^\top \mathrm{diag}\,(\boldsymbol{\rho})\, C\mathbf{x}_t$. Integrating over these coupled hidden variables and parameters is not analytically possible. However, since the model is conjugate-exponential we can apply theorem 2.2 to derive a vari-

ational Bayesian EM algorithm for state-space models analogous to the maximum-likelihood EM algorithm of Shumway and Stoffer (1982).

5.3 The variational treatment

This section covers the derivation of the results for the variational Bayesian treatment of linear-Gaussian state-space models. We first derive the lower bound on the marginal likelihood, using only the usual approximation of the factorisation of the hidden state sequence from the parameters. Due to some resulting conditional independencies between the parameters of the model, we see how the approximate posterior over parameters can be separated into posteriors for the dynamics and output processes. In section 5.3.1 the VBM step is derived, yielding approximate distributions over all the parameters of the model, each of which is analytically manageable and can be used in the VBE step.

In section 5.3.2 we justify the use of existing propagation algorithms for the VBE step, and the following subsections derive in some detail the forward and backward recursions for the variational Bayesian linear dynamical system. This section is concluded with results for hyper-parameter optimisation and a note on the tractability of the calculation of the lower bound for this model.

The variational approximation and lower bound

The full joint probability for parameters, hidden variables and observed data, given the inputs is

$$p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}, \mathbf{y}_{1:T} \,|\, \mathbf{u}_{1:T}) \;, \tag{5.23}$$

which written fully is

$$\begin{aligned} &p(A \,|\, \boldsymbol{\alpha}) p(B \,|\, \boldsymbol{\beta}) p(\boldsymbol{\rho} \,|\, a, b) p(C \,|\, \boldsymbol{\rho}, \boldsymbol{\gamma}) p(D \,|\, \boldsymbol{\rho}, \boldsymbol{\delta}) \cdot \\ &\qquad p(\mathbf{x}_0 \,|\, \boldsymbol{\mu}_0, \Sigma_0) \prod_{t=1}^{T} p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}, A, B, \mathbf{u}_t) p(\mathbf{y}_t \,|\, \mathbf{x}_t, C, D, \boldsymbol{\rho}, \mathbf{u}_t) \;. \end{aligned} \tag{5.24}$$

From this point on we drop the dependence on the input sequence $\mathbf{u}_{1:T}$, and leave it implicit. By applying Jensen's inequality we introduce any distribution $q(\boldsymbol{\theta}, \mathbf{x})$ over the parameters and hidden variables, and lower bound the log marginal likelihood

$$\ln p(\mathbf{y}_{1:T}) = \ln \int dA\, dB\, dC\, dD\, d\boldsymbol{\rho}\, d\mathbf{x}_{0:T}\; p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}, \mathbf{y}_{1:T}) \tag{5.25}$$

$$\geq \int dA\, dB\, dC\, dD\, d\boldsymbol{\rho}\, d\mathbf{x}_{0:T} \cdot q(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}) \ln \frac{p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}, \mathbf{y}_{1:T})}{q(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T})} \tag{5.26}$$

$$= \mathcal{F} \,.$$

The next step in the variational approximation is to assume some approximate form for the distribution $q(\cdot)$ which leads to a tractable bound. First, we factorise the parameters from the hidden variables giving $q(A, B, C, D, \rho, \mathbf{x}_{0:T}) = q_{\boldsymbol{\theta}}(A, B, C, D, \rho) q_{\mathbf{x}}(\mathbf{x}_{0:T})$. Writing out the expression for the exact log posterior $\ln p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{1:T}, \mathbf{y}_{0:T})$, one sees that it contains interaction terms between $\boldsymbol{\rho}$, C and D but none between $\{A, B\}$ and any of $\{\boldsymbol{\rho}, C, D\}$. This observation implies a further factorisation of the posterior parameter distributions,

$$q(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}) = q_{AB}(A, B) q_{CD\boldsymbol{\rho}}(C, D, \boldsymbol{\rho}) q_{\mathbf{x}}(\mathbf{x}_{0:T}) \,. \tag{5.27}$$

It is important to stress that this latter factorisation amongst the parameters falls out of the initial factorisation of hidden variables from parameters, and from the *resulting* conditional independencies given the hidden variables. Therefore the variational approximation does not concede any accuracy by the latter factorisation, since it is exact given the first factorisation of the parameters from hidden variables.

We choose to write the factors involved in this joint parameter distribution as

$$q_{AB}(A, B) = q_B(B)\, q_A(A \,|\, B) \tag{5.28}$$

$$q_{CD\boldsymbol{\rho}}(C, D, \boldsymbol{\rho}) = q_{\boldsymbol{\rho}}(\boldsymbol{\rho})\, q_D(D \,|\, \boldsymbol{\rho})\, q_C(C \,|\, D, \boldsymbol{\rho}) \,. \tag{5.29}$$

Now the form for $q(\cdot)$ in (5.27) causes the integral (5.26) to separate into the following sum of terms:

$$\begin{aligned}
\mathcal{F} = &\int dB\; q_B(B) \ln \frac{p(B \,|\, \boldsymbol{\beta})}{q_B(B)} + \int dB\; q_B(B) \int dA\; q_A(A \,|\, B) \ln \frac{p(A \,|\, \boldsymbol{\alpha})}{q_A(A \,|\, B)} \\
&+ \int d\boldsymbol{\rho}\; q_{\boldsymbol{\rho}}(\boldsymbol{\rho}) \ln \frac{p(\boldsymbol{\rho} \,|\, a, b)}{q_{\boldsymbol{\rho}}(\boldsymbol{\rho})} + \int d\boldsymbol{\rho}\; q_{\boldsymbol{\rho}}(\boldsymbol{\rho}) \int dD\; q_D(D \,|\, \boldsymbol{\rho}) \ln \frac{p(D \,|\, \boldsymbol{\rho}, \boldsymbol{\delta})}{q_D(D \,|\, \boldsymbol{\rho})} \\
&+ \int d\boldsymbol{\rho}\; q_{\boldsymbol{\rho}}(\boldsymbol{\rho}) \int dD\; q_D(D \,|\, \boldsymbol{\rho}) \int dC\; q_C(C \,|\, \boldsymbol{\rho}, D) \ln \frac{p(C \,|\, \boldsymbol{\rho}, \boldsymbol{\gamma})}{q_C(C \,|\, \boldsymbol{\rho}, D)} \\
&- \int d\mathbf{x}_{0:T}\; q_{\mathbf{x}}(\mathbf{x}_{0:T}) \ln q_{\mathbf{x}}(\mathbf{x}_{0:T}) \\
&+ \int dB\; q_B(B) \int dA\; q_A(A \,|\, B) \int d\boldsymbol{\rho}\; q_{\boldsymbol{\rho}}(\boldsymbol{\rho}) \int dD\; q_D(D \,|\, \boldsymbol{\rho}) \int dC\; q_C(C \,|\, \boldsymbol{\rho}, D) \cdot \\
&\qquad \int d\mathbf{x}_{0:T}\; q_{\mathbf{x}}(\mathbf{x}_{0:T}) \ln p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \,|\, A, B, C, D, \boldsymbol{\rho}) \qquad (5.30) \\
= &\mathcal{F}(q_{\mathbf{x}}(\mathbf{x}_{0:T}), q_B(B), q_A(A \,|\, B), q_{\boldsymbol{\rho}}(\boldsymbol{\rho}), q_D(D \,|\, \boldsymbol{\rho}), q_C(C \,|\, \boldsymbol{\rho}, D)) \,. \qquad (5.31)
\end{aligned}$$

Here we have left implicit the dependence of $\mathcal{F}$ on the hyperparameters. For variational Bayesian learning, $\mathcal{F}$ is the key quantity that we work with. Learning proceeds with iterative updates of the variational posterior distributions $q_\cdot(\cdot)$, each locally maximising $\mathcal{F}$.

The optimum forms of these approximate posteriors can be found by taking functional derivatives of $\mathcal{F}$ (5.30) with respect to each distribution over parameters and hidden variable sequences. In the following subsections we describe the straightforward VBM step, and the somewhat more complicated VBE step. We do not need to be able to compute $\mathcal{F}$ to produce the learning rules, only calculate its derivatives. Nevertheless its calculation at each iteration can be helpful to ensure that we are monotonically increasing a lower bound on the marginal likelihood. We finish this section on the topic of how to calculate $\mathcal{F}$ which is hard to compute because it contains the a term which is the entropy of the posterior distribution over hidden state sequences,

$$H(q_{\mathbf{x}}(\mathbf{x}_{0:T})) = -\int d\mathbf{x}_{0:T}\; q_{\mathbf{x}}(\mathbf{x}_{0:T}) \ln q_{\mathbf{x}}(\mathbf{x}_{0:T}) \,. \qquad (5.32)$$

5.3.1 VBM step: Parameter distributions

Starting from some arbitrary distribution over the hidden variables, the VBM step obtained by applying theorem 2.2 finds the variational posterior distributions over the parameters, and from these computes the expected natural parameter vector, $\overline{\boldsymbol{\phi}} = \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle$, where the expectation is taken under the distribution $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (A, B, C, D, \boldsymbol{\rho})$.

We omit the details of the derivations, and present just the forms of the distributions that extremise $\mathcal{F}$. As was mentioned in section 5.2.2, given the approximating factorisation of the

posterior distribution over hidden variables and parameters, the approximate posterior over the parameters can be factorised without further assumption or approximation into

$$q_{\boldsymbol{\theta}}(A, B, C, D, \boldsymbol{\rho}) = \prod_{j=1}^{k} q(\mathbf{b}_{(j)}) q(\mathbf{a}_{(j)} \,|\, \mathbf{b}_{(j)}) \prod_{s=1}^{p} q(\rho_s) q(\mathbf{d}_{(s)} \,|\, \rho_s) q(\mathbf{c}_{(s)} \,|\, \rho_s, \mathbf{d}_{(s)}) \tag{5.33}$$

where, for example, the row vector $\mathbf{b}_{(j)}^{\top}$ is used to denote the jth row of the matrix B (similarly so for the other parameter matrices).

We begin by defining some statistics of the input and observation data:

$$\ddot{U} \equiv \sum_{t=1}^{T} \mathbf{u}_t \mathbf{u}_t^{\top} \,, \qquad U_Y \equiv \sum_{t=1}^{T} \mathbf{u}_t \mathbf{y}_t^{\top} \,, \qquad \ddot{Y} \equiv \sum_{t=1}^{T} \mathbf{y}_t \mathbf{y}_t^{\top} \,. \tag{5.34}$$

In the forms of the variational posteriors given below, the matrix quantities W_A, G_A, $\tilde{M}$, S_A, and W_C, G_C, S_C are exactly the expected complete data sufficient statistics, obtained in the VBE step — their forms are given in equations (5.126-5.132).

The natural factorisation of the variational posterior over parameters yields these forms for A and B:

$$q_B(B) = \prod_{j=1}^{k} \mathrm{N}\left(\mathbf{b}_{(j)} \,|\, \Sigma_B \overline{\mathbf{b}}_{(j)},\, \Sigma_B\right) \tag{5.35}$$

$$q_A(A \,|\, B) = \prod_{j=1}^{k} \mathrm{N}\left(\mathbf{a}_{(j)} \,|\, \Sigma_A \left[\mathbf{s}_{A,(j)} - G_A \mathbf{b}_{(j)}\right],\, \Sigma_A\right) \tag{5.36}$$

with

$$\Sigma_A{}^{-1} = \mathrm{diag}\,(\boldsymbol{\alpha}) + W_A \tag{5.37}$$

$$\Sigma_B{}^{-1} = \mathrm{diag}\,(\boldsymbol{\beta}) + \ddot{U} - G_A^{\top} \Sigma_A G_A \tag{5.38}$$

$$\overline{B} = \tilde{M}^{\top} - S_A^{\top} \Sigma_A G_A \,, \tag{5.39}$$

and where $\overline{\mathbf{b}}_{(j)}^{\top}$ and $\mathbf{s}_{A,(j)}$ are vectors used to denote the jth row of $\overline{B}$ and the jth column of S_A respectively. It is straightforward to show that the marginal for A is given by:

$$q_A(A) = \prod_{j=1}^{k} \mathrm{N}\left(\mathbf{a}_{(j)} \,|\, \Sigma_A \left[\mathbf{s}_{A,(j)} - G_A \Sigma_B \overline{\mathbf{b}}_{(j)}\right],\, \hat{\Sigma}_A\right) \,, \tag{5.40}$$

$$\text{where} \quad \hat{\Sigma}_A = \Sigma_A + \Sigma_A G_A \Sigma_B G_A^{\top} \Sigma_A \,. \tag{5.41}$$

In the case of either the A and B matrices, for both the marginal and conditional distributions, each row has the same covariance.

The variational posterior over ρ, C and D is given by:

$$q_{\boldsymbol{\rho}}(\boldsymbol{\rho}) = \prod_{s=1}^{p} \mathrm{Ga}\left(\rho_s \,|\, a + \frac{T}{2},\ b + \frac{1}{2} G_{ss}\right) \tag{5.42}$$

$$q_D(D \,|\, \boldsymbol{\rho}) = \prod_{s=1}^{p} \mathrm{N}\left(\mathbf{d}_{(s)} \,|\, \Sigma_D \overline{\mathbf{d}}_{(s)},\ \rho_s^{-1} \Sigma_D\right) \tag{5.43}$$

$$q_C(C \,|\, D, \boldsymbol{\rho}) = \prod_{s=1}^{p} \mathrm{N}\left(\mathbf{c}_{(s)} \,|\, \Sigma_C \left[\mathbf{s}_{C,(s)} - G_C \mathbf{d}_{(s)}\right],\ \rho_s^{-1} \Sigma_C\right) \tag{5.44}$$

with

$$\Sigma_C{}^{-1} = \mathrm{diag}\left(\boldsymbol{\gamma}\right) + W_C \tag{5.45}$$

$$\Sigma_D{}^{-1} = \mathrm{diag}\left(\boldsymbol{\delta}\right) + \ddot{U} - G_C^{\top} \Sigma_C G_C \tag{5.46}$$

$$G = \ddot{Y} - S_C^{\top} \Sigma_C S_C - \overline{D} \Sigma_D \overline{D}^{\top} \tag{5.47}$$

$$\overline{D} = U_Y^{\top} - S_C^{\top} \Sigma_C G_C \ , \tag{5.48}$$

and where $\overline{\mathbf{d}}_{(s)}^{\top}$ and $\mathbf{s}_{C,(s)}$ are vectors corresponding to the sth row of $\overline{D}$ and the sth column of S_C respectively. Unlike the case of the A and B matrices, the covariances for each row of the C and D matrices can be very different due to the appearance of the ρ_s term, as so they should be. Again it is straightforward to show that the marginal for C given $\boldsymbol{\rho}$, is given by:

$$q_C(C \,|\, \boldsymbol{\rho}) = \prod_{s=1}^{p} \mathrm{N}\left(\mathbf{c}_{(s)} \,|\, \Sigma_C \left[\mathbf{s}_{C,(s)} - G_C \Sigma_D \overline{\mathbf{d}}_{(s)}\right],\ \rho_s^{-1} \hat{\Sigma}_C\right) \ , \tag{5.49}$$

$$\text{where} \quad \hat{\Sigma}_C = \Sigma_C + \Sigma_C G_C \Sigma_D G_C^{\top} \Sigma_C \ . \tag{5.50}$$

Lastly, the full marginals for C and D after integrating out the precision $\boldsymbol{\rho}$ are Student-t distributions.

In the VBM step we need to calculate the expected natural parameters, $\overline{\boldsymbol{\phi}}$, as mentioned in theorem 2.2. These will then be used in the VBE step which infers the distribution $q_{\mathbf{x}}(\mathbf{x}_{0:T})$ over hidden states in the system. The relevant natural parameterisation is given by the following:

$$\boldsymbol{\phi}(\boldsymbol{\theta}) = \boldsymbol{\phi}(A, B, C, D, R) = \Big[A,\ A^{\top}A,\ B,\ A^{\top}B,\ C^{\top}R^{-1}C,\ R^{-1}C,\ C^{\top}R^{-1}D$$
$$B^{\top}B,\ R^{-1},\ \ln\left|R^{-1}\right|,\ D^{\top}R^{-1}D,\ R^{-1}D\Big] \ . \tag{5.51}$$

The terms in the expected natural parameter vector $\overline{\boldsymbol{\phi}} = \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}$, where $\langle \cdot \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}$ denotes expectation with respect to the variational posterior, are then given by:

$$\langle A \rangle = \left[S_A - G_A \Sigma_B \overline{B}^\top \right]^\top \Sigma_A \tag{5.52}$$

$$\langle A^\top A \rangle = \langle A \rangle^\top \langle A \rangle + k \left[\Sigma_A + \Sigma_A G_A \Sigma_B G_A^\top \Sigma_A \right] \tag{5.53}$$

$$\langle B \rangle = \overline{B} \Sigma_B \tag{5.54}$$

$$\langle A^\top B \rangle = \Sigma_A \left[S_A \langle B \rangle - G_A \left\{ \langle B \rangle^\top \langle B \rangle + k \Sigma_B \right\} \right] \tag{5.55}$$

$$\langle B^\top B \rangle = \langle B \rangle^\top \langle B \rangle + k \Sigma_B \ , \tag{5.56}$$

and

$$\langle \rho_s \rangle = \overline{\rho_s} = \frac{a_{\boldsymbol{\rho}} + T/2}{b_{\boldsymbol{\rho}} + G_{ss}/2} \tag{5.57}$$

$$\langle \ln \rho_s \rangle = \overline{\ln \rho_s} = \psi(a_{\boldsymbol{\rho}} + T/2) - \ln(b_{\boldsymbol{\rho}} + G_{ss}/2) \tag{5.58}$$

$$\langle R^{-1} \rangle = \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \ , \tag{5.59}$$

$$\tag{5.60}$$

and

$$\langle C \rangle = \left[S_C - G_C \Sigma_D \overline{D}^\top \right]^\top \Sigma_C \tag{5.61}$$

$$\langle D \rangle = \overline{D} \Sigma_D \tag{5.62}$$

$$\langle C^\top R^{-1} C \rangle = \langle C \rangle^\top \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \langle C \rangle + p \left[\Sigma_C + \Sigma_C G_C \Sigma_D G_C^\top \Sigma_C \right] \tag{5.63}$$

$$\langle R^{-1} C \rangle = \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \langle C \rangle \tag{5.64}$$

$$\langle C^\top R^{-1} D \rangle = \Sigma_C \left[S_C \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \langle D \rangle - G_C \langle D \rangle^\top \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \langle D \rangle - p G_C \Sigma_D \right] \tag{5.65}$$

$$\langle R^{-1} D \rangle = \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \langle D \rangle \tag{5.66}$$

$$\langle D^\top R^{-1} D \rangle = \langle D \rangle^\top \mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right) \langle D \rangle + p \Sigma_D \ . \tag{5.67}$$

Also included in this list are several expectations which are not part of the mean natural parameter vector, but are given here because having them at hand during and after an optimisation is useful.

5.3.2 VBE step: The Variational Kalman Smoother

We now turn to the VBE step: computing $q_{\mathbf{x}}(\mathbf{x}_{0:T})$. Since SSMs are singly connected belief networks corollary 2.2 tells us that we can make use of belief propagation, which in the case of SSMs is known as the Rauch-Tung-Striebel smoother (Rauch et al., 1963). Unfortunately the

implementations of the filter and smoother are not as straightforward as one might expect, as is explained in the following subsections.

In the standard point-parameter linear-Gaussian dynamical system, given the settings of the parameters, the hidden state posterior is jointly Gaussian over the time steps. Reassuringly, when we differentiate $\mathcal{F}$ with respect to $q_{\mathbf{x}}(\mathbf{x}_{0:T})$, the variational posterior for $\mathbf{x}_{0:T}$ is also Gaussian:

$$\ln q_{\mathbf{x}}(\mathbf{x}_{0:T}) = -\ln Z + \langle \ln p(A, B, C, D, \boldsymbol{\rho}, \mathbf{x}_{0:T}, \mathbf{y}_{1:T}) \rangle \tag{5.68}$$

$$= -\ln Z' + \langle \ln p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \,|\, A, B, C, D, \boldsymbol{\rho}) \rangle \;, \tag{5.69}$$

where

$$Z' = \int d\mathbf{x}_{0:T} \;\; \exp \langle \ln p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \,|\, A, B, C, D, \boldsymbol{\rho}) \rangle \;, \tag{5.70}$$

and where $\langle \cdot \rangle$ denotes expectation with respect to the variational posterior distribution over parameters, $q_{\boldsymbol{\theta}}(A, B, C, D, \boldsymbol{\rho})$. In this expression the expectations with respect to the approximate parameter posteriors are performed on the logarithm of the complete-data likelihood and, even though this leaves the coefficients on the $\mathbf{x}_t$ terms in a somewhat unorthodox state, the new log posterior still only contains up to quadratic terms in each $\mathbf{x}_t$ and therefore $q_{\mathbf{x}}(\mathbf{x}_{0:T})$ must be Gaussian, as in the point-parameter case. We should therefore still be able to use an algorithm very similar to the Kalman filter and smoother for inference of the hidden state sequence's sufficient statistics (the E-like step). However we can no longer plug in parameters to the filter and smoother, but have to work with the natural parameters throughout the implementation.

The following paragraphs take us through the required derivations for the forward and backward recursions. For the sake of clarity of exposition, we do not at this point derive the algorithms for the input-driven system (though we do present the full input-driven algorithms as pseudocode in algorithms 5.1, 5.2 and 5.3). At each stage, we first we concentrate on the point-parameter propagation algorithms and then formulate the Bayesian analogues.

5.3.3 Filter (forward recursion)

In this subsection, we first derive the well-known forward filtering recursion steps for the case in which the parameters are fixed point-estimates. The variational Bayesian analogue of the forward pass is then presented. The dependence of the filter equations on the inputs $\mathbf{u}_{1:T}$ has been omitted in the derivations, but is included in the summarising algorithms.

Point-parameter derivation

We define $\alpha_t(\mathbf{x}_t)$ to be the posterior over the hidden state at time t given observed data up to and including time t:

$$\alpha_t(\mathbf{x}_t) \equiv p(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) \,. \tag{5.71}$$

Note that this is slightly different to the traditional form for HMMs which is $\alpha_t(\mathbf{x}_t) \equiv p(\mathbf{x}_t, \mathbf{y}_{1:t})$. We then form the recursion with $\alpha_{t-1}(\mathbf{x}_{t-1})$ as follows:

$$\begin{aligned}
\alpha_t(\mathbf{x}_t) &= \int d\mathbf{x}_{t-1}\; p(\mathbf{x}_{t-1} \,|\, \mathbf{y}_{1:t-1})\, p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\, p(\mathbf{y}_t \,|\, \mathbf{x}_t) \;/\; p(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}) && (5.72)\\
&= \frac{1}{\zeta_t(\mathbf{y}_t)} \int d\mathbf{x}_{t-1}\; \alpha_{t-1}(\mathbf{x}_{t-1})\, p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})\, p(\mathbf{y}_t \,|\, \mathbf{x}_t) && (5.73)\\
&= \frac{1}{\zeta_t(\mathbf{y}_t)} \int d\mathbf{x}_{t-1}\; \mathrm{N}(\mathbf{x}_{t-1} \,|\, \boldsymbol{\mu}_{t-1}, \Sigma_{t-1})\; \mathrm{N}(\mathbf{x}_t \,|\; A\mathbf{x}_{t-1}, \mathrm{I})\; \mathrm{N}(\mathbf{y}_t \,|\; C\mathbf{x}_t, R) && (5.74)\\
&= \mathrm{N}(\mathbf{x}_t \,|\; \boldsymbol{\mu}_t, \Sigma_t) && (5.75)
\end{aligned}$$

where

$$\zeta_t(\mathbf{y}_t) \equiv p(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}) \tag{5.76}$$

is the filtered output probability; this will be useful for computing the likelihood. Within the above integrand the quadratic terms in $\mathbf{x}_{t-1}$ form the Gaussian $\mathrm{N}(\mathbf{x}_{t-1} \,|\, \mathbf{x}^*_{t-1}, \Sigma^*_{t-1})$ with

$$\begin{aligned}
\Sigma^*_{t-1} &= \left(\Sigma^{-1}_{t-1} + A^\top A\right)^{-1} && (5.77)\\
\mathbf{x}^*_{t-1} &= \Sigma^*_{t-1}\left[\Sigma^{-1}_{t-1}\boldsymbol{\mu}_{t-1} + A^\top \mathbf{x}_t\right] \,. && (5.78)
\end{aligned}$$

Marginalising out $\mathbf{x}_{t-1}$ gives the filtered estimates of the mean and covariance of the hidden state as

$$\alpha_t(\mathbf{x}_t) = \mathrm{N}(\mathbf{x}_t \,|\, \boldsymbol{\mu}_t, \Sigma_t) \tag{5.79}$$

with

$$\begin{aligned}
\Sigma_t &= \left[\mathrm{I} + C^\top R^{-1} C - A\Sigma^*_{t-1}A^\top\right]^{-1} && (5.80)\\
\boldsymbol{\mu}_t &= \Sigma_t\left[C^\top R^{-1}\mathbf{y}_t + A\Sigma^*_{t-1}\Sigma^{-1}_{t-1}\boldsymbol{\mu}_{t-1}\right] \,. && (5.81)
\end{aligned}$$

At each step the normalising constant ζ_t, obtained as the denominator in (5.72), contributes to the calculation of the probability of the data

$$\begin{aligned}
p(\mathbf{y}_{1:T}) &= p(\mathbf{y}_1)p(\mathbf{y}_2 \,|\, \mathbf{y}_1) \ldots p(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}) \ldots p(\mathbf{y}_T \,|\, \mathbf{y}_{1:T-1}) && (5.82)\\
&= p(\mathbf{y}_1) \prod_{t=2}^{T} p(\mathbf{y}_t \,|\, \mathbf{y}_{1:t-1}) = \prod_{t=1}^{T} \zeta_t(\mathbf{y}_t) \,. && (5.83)
\end{aligned}$$

It is not difficult to show that each of the above terms are Gaussian distributed,

$$\zeta_t(\mathbf{y}_t) = \mathrm{N}(\mathbf{y}_t \,|\, \boldsymbol{\varpi}_t, \varsigma_t) \tag{5.84}$$

with

$$\varsigma_t = \left(R^{-1} - R^{-1}C\Sigma_t C^\top R^{-1}\right)^{-1} \tag{5.85}$$

$$\boldsymbol{\varpi}_t = \varsigma_t R^{-1} C \Sigma_t A \Sigma^*_{t-1} \Sigma^{-1}_{t-1} \boldsymbol{\mu}_{t-1} \,. \tag{5.86}$$

With these distributions at hand we can compute the probability of each observation $\mathbf{y}_t$ given the previous observations in the sequence, and assign a predictive mean and variance to the data at each time step as it arrives. However, this predictive distribution will change once the hidden state sequence has been smoothed on the backward pass.

Certain expressions such as equations (5.80), (5.81), and (5.85) could be simplified using the matrix inversion lemma (see appendix B.2), but here we refrain from doing so because a similar operation is not possible in the variational Bayesian derivation (see comment at end of section 5.3.3).

Variational derivation

It is quite straightforward to repeat the above derivation for variational Bayesian learning, by replacing parameters (and combinations of parameters) with their expectations under the variational posterior distributions which were calculated in the VBM step (section 5.3.1). Equation (5.74) becomes rewritten as

$$\begin{aligned}
\alpha_t(\mathbf{x}_t) = \frac{1}{\zeta'_t(\mathbf{y}_t)} \int d\mathbf{x}_{t-1}\, \mathrm{N}(\mathbf{x}_{t-1} \,|\, \boldsymbol{\mu}_{t-1}, \Sigma_{t-1}) \cdot \\
\exp -\frac{1}{2}\Big\langle (\mathbf{x}_t - A\mathbf{x}_{t-1})^\top \mathrm{I} (\mathbf{x}_t - A\mathbf{x}_{t-1}) + (\mathbf{y}_t - C\mathbf{x}_t)^\top R^{-1} (\mathbf{y}_t - C\mathbf{x}_t) \\
+ k \ln |2\pi| + \ln |2\pi R| \Big\rangle \qquad (5.87) \\
= \frac{1}{\zeta'_t(\mathbf{y}_t)} \int d\mathbf{x}_{t-1}\, \mathrm{N}(\mathbf{x}_{t-1} \,|\, \boldsymbol{\mu}_{t-1}, \Sigma_{t-1}) \cdot \\
\exp -\frac{1}{2}\Big[\mathbf{x}_{t-1}^\top \langle A^\top A \rangle \mathbf{x}_{t-1} - 2\mathbf{x}_{t-1}^\top \langle A \rangle^\top \mathbf{x}_t \\
+ \mathbf{x}_t^\top (\mathrm{I} + \langle C^\top R^{-1} C \rangle) \mathbf{x}_t - 2\mathbf{x}_t^\top \langle C^\top R^{-1} \rangle \mathbf{y}_t + \ldots \Big] \qquad (5.88)
\end{aligned}$$

where the angled brackets $\langle \cdot \rangle$ denote expectation under the variational posterior distribution over parameters, $q_{\boldsymbol{\theta}}(A, B, C, D, \boldsymbol{\rho})$.

After the parameter averaging, the integrand is still log-quadratic in both $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$, and so the derivation continues as before but with parameter expectations taking place of the point estimates. Equations (5.77) and (5.78) now become

$$\Sigma^*_{t-1} = \left(\Sigma^{-1}_{t-1} + \langle A^\top A \rangle\right)^{-1} \tag{5.89}$$

$$\mathbf{x}^*_{t-1} = \Sigma^*_{t-1}\left[\Sigma^{-1}_{t-1}\boldsymbol{\mu}_{t-1} + \langle A \rangle^\top \mathbf{x}_t\right] , \tag{5.90}$$

and marginalising out $\mathbf{x}_{t-1}$ yields a Gaussian distribution over $\mathbf{x}_t$,

$$\alpha_t(\mathbf{x}_t) = \mathrm{N}(\mathbf{x}_t \mid \boldsymbol{\mu}_t, \Sigma_t) \tag{5.91}$$

with mean and covariance given by

$$\Sigma_t = \left[\mathrm{I} + \langle C^\top R^{-1} C \rangle - \langle A \rangle \Sigma^*_{t-1} \langle A \rangle^\top\right]^{-1} \tag{5.92}$$

$$\boldsymbol{\mu}_t = \Sigma_t\left[\langle C^\top R^{-1} \rangle \mathbf{y}_t + \langle A \rangle \Sigma^*_{t-1}\Sigma^{-1}_{t-1}\boldsymbol{\mu}_{t-1}\right] . \tag{5.93}$$

This variational α-message evidently resembles the point-parameter result in (5.80) and (5.81). Algorithm 5.1 shows the full implementation for the variational Bayesian forward recursion, including extra terms from the inputs and input-related parameters B and D which were not derived here to keep the presentation concise. In addition it gives the variational Bayesian analogues of equations (5.85) and (5.86).

We now see why, for example, equation (5.85) was not simplified using the matrix inversion lemma — this operation would necessarily split the R^{-1} and C matrices, yet its variational Bayesian counterpart requires that expectations be taken over the combined product $R^{-1}C$. These expectations cannot be passed through the inversion lemma. Included in appendix B.2 is a proof of the matrix inversion lemma which shows clearly how such expectations would become disjoined.

5.3.4 Backward recursion: sequential and parallel

In the backward pass information about future observations is incorporated to update the posterior distribution on the current time step. This recursion begins at the last time step $t = T$ (which has no future observations to take into account) and recurses to the beginning of the sequence to time $t = 0$.

There are two different forms for the backward pass. The *sequential* form makes use of the α-messages from the forward pass and does not need to access information about the current observation in order to calculate the posterior over the hidden state given all the data. The *parallel* form is so-called because it executes all its recursions independently of the forward

Algorithm 5.1: Forward recursion for variational Bayesian state-space models with inputs $\mathbf{u}_{1:T}$ (variational Kalman filter).

---

1. Initialise hyperparameters $\boldsymbol{\mu}_0$ and Σ_0 as the mean and covariance of the auxiliary hidden state $\mathbf{x}_0$

2. For $t = 1$ to T

   (a) Compute $\alpha_t(\mathbf{x}_t) = \mathrm{N}(\mathbf{x}_t \,|\, \boldsymbol{\mu}_t, \Sigma_t)$

$$\begin{aligned}
\Sigma^*_{t-1} &= \left(\Sigma^{-1}_{t-1} + \langle A^\top A \rangle\right)^{-1} \\
\Sigma_t &= \left(I + \langle C^\top R^{-1} C \rangle - \langle A \rangle \Sigma^*_{t-1} \langle A \rangle^\top\right)^{-1} \\
\boldsymbol{\mu}_t &= \Sigma_t \Big[\langle C^\top R^{-1} \rangle \mathbf{y}_t + \langle A \rangle \Sigma^*_{t-1} \Sigma^{-1}_{t-1} \boldsymbol{\mu}_{t-1} \\
&\qquad + \left(\langle B \rangle - \langle A \rangle \Sigma^*_{t-1} \langle A^\top B \rangle - \langle C^\top R^{-1} D \rangle\right) \mathbf{u}_t\Big]
\end{aligned}$$

   (b) Compute predictive distribution of $\mathbf{y}_t$

$$\begin{aligned}
\varsigma_t &= \left(\langle R^{-1} \rangle - \langle R^{-1} C \rangle \Sigma_t \langle R^{-1} C \rangle^\top\right)^{-1} \\
\boldsymbol{\varpi}_t &= \varsigma_t \Big[\langle R^{-1} C \rangle \Sigma_t \langle A \rangle \Sigma^*_{t-1} \Sigma^{-1}_{t-1} \boldsymbol{\mu}_{t-1} \\
&\qquad + \left(\langle R^{-1} D \rangle + \langle R^{-1} C \rangle \Sigma_t \left\{\langle B \rangle - \langle C^\top R^{-1} D \rangle - \langle A \rangle \Sigma^*_{t-1} \langle A^\top B \rangle\right\}\right) \mathbf{u}_t\Big]
\end{aligned}$$

   (c) Compute $\zeta'_t(\mathbf{y}_t)$ (see (5.87) and also section 5.3.7 for details)

$$\begin{aligned}
\ln \zeta'_t(\mathbf{y}_t) = -\frac{1}{2}\Big[&\langle \ln |2\pi R| \rangle - \ln \left|\Sigma^{-1}_{t-1} \Sigma^*_{t-1} \Sigma_t\right| + \boldsymbol{\mu}^\top_{t-1} \Sigma^{-1}_{t-1} \boldsymbol{\mu}_{t-1} - \boldsymbol{\mu}^\top_t \Sigma^{-1}_t \boldsymbol{\mu}_t \\
&+ \mathbf{y}^\top_t \langle R^{-1} \rangle \mathbf{y}_t - 2\mathbf{y}^\top_t \langle R^{-1} D \rangle \mathbf{u}_t + \mathbf{u}^\top_t \langle D^\top R^{-1} D \rangle \mathbf{u}_t \\
&- (\Sigma^{-1}_{t-1} \boldsymbol{\mu}_{t-1} - \langle A^\top B \rangle \mathbf{u}_t)^\top \Sigma^*_{t-1} (\Sigma^{-1}_{t-1} \boldsymbol{\mu}_{t-1} - \langle A^\top B \rangle \mathbf{u}_t)\Big]
\end{aligned}$$

   End For

3. Output all computed quantities, including

   $\ln Z' = \sum_{t=1}^{T} \ln \zeta'_t(\mathbf{y}_t)$

pass, and then later combines its messages with those from the forward pass to compute the hidden state posterior for each time step.

Sequential implementation: point-parameters

In the sequential implementation we define a set of γ-messages to be the posterior over the hidden state given all the data. In the case of point-parameters, the recursion is then

$$\begin{aligned}
\gamma_t(\mathbf{x}_t) &\equiv p(\mathbf{x}_t \,|\, \mathbf{y}_{1:T}) && (5.94)\\
&= \int d\mathbf{x}_{t+1}\; p(\mathbf{x}_t, \mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:T}) && (5.95)\\
&= \int d\mathbf{x}_{t+1}\; p(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:T}) p(\mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:T}) && (5.96)\\
&= \int d\mathbf{x}_{t+1}\; p(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:t})\; p(\mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:T}) && (5.97)\\
&= \int d\mathbf{x}_{t+1}\; \left[\frac{p(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t)}{\int d\mathbf{x}'_t\; p(\mathbf{x}'_t \,|\, \mathbf{y}_{1:t}) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}'_t)}\right] p(\mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:T}) && (5.98)\\
&= \int d\mathbf{x}_{t+1}\; \left[\frac{\alpha_t(\mathbf{x}_t) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t)}{\int d\mathbf{x}'_t\; \alpha_t(\mathbf{x}'_t) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}'_t)}\right] \gamma_{t+1}(\mathbf{x}_{t+1})\;. && (5.99)
\end{aligned}$$

Here the use of Bayes' rule in (5.98) has had the effect of replacing the explicit data dependence with functions of the α-messages computed in the forward pass. Integrating out $\mathbf{x}_{t+1}$ yields Gaussian distributions for the smoothed estimates of the hidden state at each time step:

$$\gamma_t(\mathbf{x}_t) = \mathrm{N}(\mathbf{x}_t \,|\, \boldsymbol{\omega}_t, \Upsilon_{tt}) \tag{5.100}$$

where Σ^*_t is as defined in the forward pass according to (5.77) and

$$\begin{aligned}
K_t &= \left(\Upsilon^{-1}_{t+1,t+1} + A\Sigma^*_t A^\top\right)^{-1} && (5.101)\\
\Upsilon_{tt} &= \left[\Sigma^{*-1}_t - A^\top K_t A\right]^{-1} && (5.102)\\
\boldsymbol{\omega}_t &= \Upsilon_{tt}\left[\Sigma^{-1}_t \boldsymbol{\mu}_t + A^\top K_t \left(\Upsilon^{-1}_{t+1,t+1}\boldsymbol{\omega}_{t+1} - A\Sigma^*_t \Sigma^{-1}_t \boldsymbol{\mu}_t\right)\right]\;. && (5.103)
\end{aligned}$$

Note that K_t given in (5.101) is a different matrix to the Kalman gain matrix as found in the Kalman filtering and smoothing literature, and should not be confused with it.

The sequential version has an advantage in online scenarios: once the data at time t, $\mathbf{y}_t$, has been filtered it can be discarded and is replaced with its message, $\alpha_t(\mathbf{x}_t)$ (see, for example, Rauch, 1963). In this way potentially high dimensional observations can be stored simply as beliefs in the low dimensional state space.

Sequential implementation: variational analysis

Unfortunately the step using Bayes' rule in (5.98) cannot be transferred over to a variational treatment, and this can be demonstrated by seeing how the term $p(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:t})$ in (5.97) is altered by the lower bound operation. Up to a normalisation factor,

$$p(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:t}) \overset{\text{VB}}{\to} \exp\Big\langle \ln p(\mathbf{x}_t \,|\, \mathbf{x}_{t+1}, \mathbf{y}_{1:t}) \Big\rangle \tag{5.104}$$

$$= \exp\Big\langle \ln p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t) + \ln \alpha_t(\mathbf{x}_t) - \ln \int d\mathbf{x}'_t \; \alpha_t(\mathbf{x}'_t) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}'_t) \Big\rangle \tag{5.105}$$

The last term in the above equation results in a precision term in the exponent of the form: $\ln \int d\mathbf{x}'_t \; \alpha_t(\mathbf{x}'_t) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}'_t) = -\frac{1}{2}\left[\mathrm{I} - A\left[\Sigma_t^{-1} + A^\top A\right]^{-1} A^\top\right] + c$. Even though this term is easy to express for a known A matrix, its expectation under $q_A(A)$ is difficult to compute. Even with the use of the matrix inversion lemma (see appendix B.2), which yields $\left(\mathrm{I} + A\Sigma_t A^\top\right)^{-1}$, the expression is still not amenable to expectation.

Parallel implementation: point-parameters

Some of the above problems are ameliorated using the parallel implementation, which we first derive using point-parameters. The parallel recursion produces β-messages, defined as

$$\beta_t(\mathbf{x}_t) \equiv p(\mathbf{y}_{t+1:T} \,|\, \mathbf{x}_t) \; . \tag{5.106}$$

These are obtained through a recursion analogous to the forward pass (5.72)

$$\beta_{t-1}(\mathbf{x}_{t-1}) = \int d\mathbf{x}_t \; p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}) p(\mathbf{y}_t \,|\, \mathbf{x}_t) p(\mathbf{y}_{t+1:T} \,|\, \mathbf{x}_t) \tag{5.107}$$

$$= \int d\mathbf{x}_t \; p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1}) p(\mathbf{y}_t \,|\, \mathbf{x}_t) \beta_t(\mathbf{x}_t) \tag{5.108}$$

$$\propto \mathrm{N}(\mathbf{x}_{t-1} \,|\, \boldsymbol{\eta}_{t-1}, \Psi_{t-1}) \tag{5.109}$$

with the end condition that $\beta_T(\mathbf{x}_T) = 1$. Omitting the details, the terms for the backward messages are given by:

$$\Psi_t^* = \left(\mathrm{I} + C^\top R^{-1} C + \Psi_t^{-1}\right)^{-1} \tag{5.110}$$

$$\Psi_{t-1} = \left[A^\top A - A^\top \Psi_t^* A\right]^{-1} \tag{5.111}$$

$$\boldsymbol{\eta}_{t-1} = \Psi_{t-1} A^\top \Psi_t^* \left[C^\top R^{-1} \mathbf{y}_t + \Psi_t^{-1} \boldsymbol{\eta}_t\right] \tag{5.112}$$

where $t = \{T, \ldots, 1\}$, and Ψ_T^{-1} set to $\mathbf{0}$ to satisfy the end condition (regardless of $\boldsymbol{\eta}_T$). The last step in this recursion therefore finds the probability of all the data given the setting of the auxiliary $\mathbf{x}_0$ variable.

Parallel implementation: variational analysis

It is straightforward to produce the variational counterpart of the backward parallel pass just described. Omitting the derivation, the results are presented in algorithm 5.2 which also includes the influence of inputs on the recursions.

Algorithm 5.2: Backward parallel recursion for variational Bayesian state-space models with inputs $\mathbf{u}_{1:T}$.

1. Initialise $\Psi_T^{-1} = \mathbf{0}$ to satisfy end condition $\beta_T(\mathbf{x}_T) = 1$
2. For $t = T$ to 1

$$\begin{aligned}
\Psi_t^* &= \left(\mathrm{I} + \langle C^\top R^{-1} C\rangle + \Psi_t^{-1}\right)^{-1} \\
\Psi_{t-1} &= \left(\langle A^\top A\rangle - \langle A\rangle^\top \Psi_t^* \langle A\rangle\right)^{-1} \\
\boldsymbol{\eta}_{t-1} &= \Psi_{t-1}\Big[-\langle A^\top B\rangle \mathbf{u}_t \\
&\qquad + \langle A\rangle^\top \Psi_t^* \left(\langle B\rangle \mathbf{u}_t + \langle C^\top R^{-1}\rangle \mathbf{y}_t - \langle C^\top R^{-1} D\rangle \mathbf{u}_t + \Psi_t^{-1}\boldsymbol{\eta}_t\right)\Big]
\end{aligned}$$

   End For

3. Output $\{\boldsymbol{\eta}_t, \Psi_t\}_{t=0}^T$

5.3.5 Computing the single and joint marginals

The culmination of the VBE step is to compute the sufficient statistics of the hidden state, which are the marginals at each time step and the pairwise marginals across adjacent time steps.

In the point-parameter case, one can use the sequential backward pass, and then the single state marginals are given exactly by the γ-messages, and it only remains to calculate the pairwise marginals. It is not difficult to show that the terms involving $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ are best represented with the quadratic term

$$\ln p(\mathbf{x}_t, \mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:T}) = -\frac{1}{2}\begin{pmatrix} \mathbf{x}_t^\top & \mathbf{x}_{t+1}^\top \end{pmatrix}\begin{pmatrix} \Sigma_t^{*-1} & -A^\top \\ -A & K_t^{-1} \end{pmatrix}\begin{pmatrix} \mathbf{x}_t \\ \mathbf{x}_{t+1} \end{pmatrix} + \text{const.}\,, \quad (5.113)$$

where Σ_t^* is computed in the forward pass (5.77) and K_t is computed in the backward sequential pass (5.101).

We define $\Upsilon_{t,t+1}$ to be the cross-covariance between the hidden states at times t and $t+1$, given all the observations $\mathbf{y}_{1:T}$:

$$\Upsilon_{t,t+1} \equiv \left\langle \left(\mathbf{x}_t - \langle \mathbf{x}_t \rangle\right) \left(\mathbf{x}_{t+1} - \langle \mathbf{x}_{t+1} \rangle\right)^\top \right\rangle , \tag{5.114}$$

where $\langle \cdot \rangle$ denotes expectation with respect to the posterior distribution over the hidden state sequence given all the data. We now make use of the Schur complements (see appendix B.1) of the precision matrix given in (5.113) to obtain

$$\Upsilon_{t,t+1} = \Sigma_t^* A^\top \Upsilon_{t+1,t+1} . \tag{5.115}$$

The variational Bayesian implementation

In the variational Bayesian scenario the marginals cannot be obtained easily with a backward sequential pass, and they are instead computed by combining the α- and β-messages as follows:

$$\begin{aligned} p(\mathbf{x}_t \,|\, \mathbf{y}_{1:T}) &\propto p(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) p(\mathbf{y}_{t+1:T} \,|\, \mathbf{x}_t) && (5.116)\\ &= \alpha_t(\mathbf{x}_t)\beta_t(\mathbf{x}_t) && (5.117)\\ &= \mathrm{N}(\mathbf{x}_t \,|\, \boldsymbol{\omega}_t, \Upsilon_{tt}) && (5.118) \end{aligned}$$

with

$$\begin{aligned} \Upsilon_{t,t} &= \left[\Sigma_t^{-1} + \Psi_t^{-1}\right]^{-1} && (5.119)\\ \boldsymbol{\omega}_t &= \Upsilon_{t,t}\left[\Sigma_t^{-1}\boldsymbol{\mu}_t + \Psi_t^{-1}\boldsymbol{\eta}_t\right] . && (5.120) \end{aligned}$$

This is computed for $t = \{0, \ldots, T-1\}$. At $t = 0$, $\alpha_0(\mathbf{x}_0)$ is exactly the prior (5.13) over the auxiliary hidden state; at $t = T$, there is no need for a calculation since $p(\mathbf{x}_T \,|\, \mathbf{y}_{1:T}) \equiv \alpha_T(\mathbf{x}_T)$.

Similarly the pairwise marginals are given by

$$\begin{aligned} p(\mathbf{x}_t, \mathbf{x}_{t+1} \,|\, \mathbf{y}_{1:T}) &\propto p(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t) p(\mathbf{y}_{t+1} \,|\, \mathbf{x}_{t+1}) p(\mathbf{y}_{t+2:T} \,|\, \mathbf{x}_{t+1}) && (5.121)\\ &= \alpha_t(\mathbf{x}_t) p(\mathbf{x}_{t+1} \,|\, \mathbf{x}_t) p(\mathbf{y}_{t+1} \,|\, \mathbf{x}_{t+1}) \beta_{t+1}(\mathbf{x}_{t+1}) , && (5.122) \end{aligned}$$

which under the variational transform becomes

$$\overset{\text{VB}}{\rightarrow} \alpha_t(\mathbf{x}_t)\ \exp\big\langle \ln p(\mathbf{x}_{t+1}\,|\,\mathbf{x}_t) + \ln p(\mathbf{y}_{t+1}\,|\,\mathbf{x}_{t+1})\big\rangle\ \beta_{t+1}(\mathbf{x}_{t+1}) \tag{5.123}$$

$$= \mathrm{N}\left(\begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t+1} \end{bmatrix} \,|\, \begin{bmatrix} \boldsymbol{\omega}_t \\ \boldsymbol{\omega}_{t+1} \end{bmatrix}, \begin{bmatrix} \Upsilon_{t,t} & \Upsilon_{t,t+1} \\ \Upsilon_{t,t+1}^\top & \Upsilon_{t+1,t+1} \end{bmatrix} \right) . \tag{5.124}$$

With the use of Schur complements again, it is not difficult to show that $\Upsilon_{t,t+1}$ is given by

$$\Upsilon_{t,t+1} = \Sigma_t^* \langle A \rangle^\top \left(\mathrm{I} + \langle C^\top R^{-1} C \rangle + \Psi_{t+1}^{-1} - \langle A \rangle \Sigma_t^* \langle A \rangle^\top \right)^{-1} . \tag{5.125}$$

This cross-covariance is then computed for all time steps $t = \{0, \ldots, T-1\}$, which includes the cross-covariance between the zeroth and first hidden states.

In summary, the entire VBE step consists of a forward pass followed by a backward pass, during which the marginals can be computed as well straight after each β-message.

The required sufficient statistics of the hidden state

In the VBE step we need to calculate the expected sufficient statistics of the hidden state, as mentioned in theorem 2.2. These will then be used in the VBM step which infers the distribution $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ over parameters of the system (section 5.3.1). The relevant expectations are:

$$W_A = \sum_{t=1}^{T} \langle \mathbf{x}_{t-1} \mathbf{x}_{t-1}^\top \rangle = \sum_{t=1}^{T} \Upsilon_{t-1,t-1} + \boldsymbol{\omega}_{t-1} \boldsymbol{\omega}_{t-1}^\top \tag{5.126}$$

$$G_A = \sum_{t=1}^{T} \langle \mathbf{x}_{t-1} \rangle \mathbf{u}_t^\top \quad = \sum_{t=1}^{T} \boldsymbol{\omega}_{t-1} \mathbf{u}_t^\top \tag{5.127}$$

$$\tilde{M} = \sum_{t=1}^{T} \mathbf{u}_t \langle \mathbf{x}_t \rangle^\top \quad = \sum_{t=1}^{T} \mathbf{u}_t \boldsymbol{\omega}_t^\top \tag{5.128}$$

$$S_A = \sum_{t=1}^{T} \langle \mathbf{x}_{t-1} \mathbf{x}_t^\top \rangle \quad = \sum_{t=1}^{T} \Upsilon_{t-1,t} + \boldsymbol{\omega}_{t-1} \boldsymbol{\omega}_t^\top \tag{5.129}$$

$$W_C = \sum_{t=1}^{T} \langle \mathbf{x}_t \mathbf{x}_t^\top \rangle = \sum_{t=1}^{T} \Upsilon_{t,t} + \boldsymbol{\omega}_t \boldsymbol{\omega}_t^\top \tag{5.130}$$

$$G_C = \sum_{t=1}^{T} \langle \mathbf{x}_t \rangle \mathbf{u}_t^\top = \sum_{t=1}^{T} \boldsymbol{\omega}_t \mathbf{u}_t^\top \tag{5.131}$$

$$S_C = \sum_{t=1}^{T} \langle \mathbf{x}_t \rangle \mathbf{y}_t^\top = \sum_{t=1}^{T} \boldsymbol{\omega}_t \mathbf{y}_t^\top . \tag{5.132}$$

Note that M and G_C are transposes of one another. Also note that all the summations contain T terms (instead of those for the dynamics model containing $T-1$). This is a consequence of our adoption of a slightly unorthodox model specification of linear dynamical systems which includes a fictitious auxiliary hidden variable $\mathbf{x}_0$.

5.3.6 Hyperparameter learning

The hyperparameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$, a and b, and the prior parameters Σ_0 and $\boldsymbol{\mu}_0$, can be updated so as to maximise the lower bound on the marginal likelihood (5.30). By taking derivatives of $\mathcal{F}$ with respect to the hyperparameters, the following updates can be derived, applicable after a VBM step:

$$\alpha_j^{-1} \leftarrow \frac{1}{k}\Big[k\Sigma_A + \Sigma_A\big[S_A S_A^\top - 2G_A\langle B\rangle^\top S_A^\top + G_A\{k\Sigma_B + \langle B\rangle^\top\langle B\rangle\}G_A^\top\big]\Sigma_A\Big]_{jj} \tag{5.133}$$

$$\beta_j^{-1} \leftarrow \frac{1}{k}\Big[k\Sigma_B + \langle B\rangle^\top\langle B\rangle\Big]_{jj} \tag{5.134}$$

$$\begin{aligned}\gamma_j^{-1} \leftarrow \frac{1}{p}\Big[&p\Sigma_C + \Sigma_C\big[S_C\mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right)S_C^\top - 2S_C\mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right)\langle D\rangle G_C^\top \\ &+ pG_C\Sigma_D G_C' + G_C\langle D\rangle^\top\mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right)\langle D\rangle G_C^\top\big]\Sigma_C\Big]_{jj}\end{aligned} \tag{5.135}$$

$$\delta_j^{-1} \leftarrow \frac{1}{p}\Big[p\Sigma_D + \langle D\rangle^\top\mathrm{diag}\left(\overline{\boldsymbol{\rho}}\right)\langle D\rangle\Big]_{jj} \tag{5.136}$$

where $[\cdot]_{jj}$ denotes its (j,j)th element.

Similarly, in order to maximise the probability of the hidden state sequence under the prior, the hyperparameters of the prior over the auxiliary hidden state are set according to the distribution of the smoothed estimate of $\mathbf{x}_0$:

$$\Sigma_0 \leftarrow \Upsilon_{0,0}\,, \qquad \boldsymbol{\mu}_0 \leftarrow \boldsymbol{\omega}_0\,. \tag{5.137}$$

Last of all, the hyperparameters a and b governing the prior distribution over the output noise, $R = \mathrm{diag}\left(\boldsymbol{\rho}\right)$, are set to the fixed point of the equations

$$\psi(a) = \ln b + \frac{1}{p}\sum_{s=1}^{p}\overline{\ln\rho_s}\,, \qquad \frac{1}{b} = \frac{1}{pa}\sum_{s=1}^{p}\overline{\rho_s} \tag{5.138}$$

where $\psi(x) \equiv \partial/\partial x \ln\Gamma(x)$ is the *digamma* function (refer to equations (5.57) and (5.58) for required expectations). These fixed point equations can be solved straightforwardly using gradient following techniques (such as Newton's method) in just a few iterations, bearing in mind the positivity constraints on a and b (see appendix C.2 for more details).

5.3.7 Calculation of $\mathcal{F}$

Before we see why $\mathcal{F}$ is hard to compute in this model, we should rewrite the lower bound more succinctly using the following definitions, in the case of a pair of variables J and K:

$$\mathrm{KL}(J) \equiv \int dJ\; q(J) \ln \frac{q(J)}{p(J)} \qquad \text{(KL divergence)} \tag{5.139}$$

$$\mathrm{KL}(J \,|\, K) \equiv \int dJ\; q(J \,|\, K) \ln \frac{q(J \,|\, K)}{p(J \,|\, K)} \qquad \text{(conditional KL)} \tag{5.140}$$

$$\langle \mathrm{KL}(J \,|\, K) \rangle_{q(K)} \equiv \int dK\; q(K) \mathrm{KL}(J \,|\, K) \qquad \text{(expected conditional KL) .} \tag{5.141}$$

Note that in (5.140) the prior over J may need to be a function of K for conjugacy reasons (this is the case for state-space models for the output parameters C and D, and the noise R). The notation $\mathrm{KL}(J \,|\, K)$ is not to be confused with $\mathrm{KL}(J||K)$ which is the KL divergence between distributions $q(J)$ and $q(K)$ (which are marginals). The lower bound $\mathcal{F}$ (5.26) can now be written as

$$\begin{aligned}
\mathcal{F} = &-\mathrm{KL}(B) - \langle \mathrm{KL}(A \,|\, B) \rangle_{q(B)} \\
&- \mathrm{KL}(\boldsymbol{\rho}) - \langle \mathrm{KL}(D \,|\, \boldsymbol{\rho}) \rangle_{q(\boldsymbol{\rho})} - \langle \mathrm{KL}(C \,|\, \boldsymbol{\rho}, D) \rangle_{q(\boldsymbol{\rho}, D)} \\
&+ \mathrm{H}(q_{\mathbf{x}}(\mathbf{x}_{0:T})) \\
&+ \langle \ln p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T} \,|\, A, B, C, D, \boldsymbol{\rho}) \rangle_{q(A,B,C,D,\boldsymbol{\rho}) q(\mathbf{x}_{1:T})}
\end{aligned} \tag{5.142}$$

where $\mathrm{H}(q_{\mathbf{x}}(\mathbf{x}_{0:T}))$ is the entropy of the variational posterior over the hidden state sequence,

$$\mathrm{H}(q_{\mathbf{x}}(\mathbf{x}_{0:T})) \equiv -\int d\mathbf{x}_{0:T}\; q_{\mathbf{x}}(\mathbf{x}_{0:T}) \ln q_{\mathbf{x}}(\mathbf{x}_{0:T}) \; . \tag{5.143}$$

The reason why $\mathcal{F}$ can not be computed directly is precisely due to both this entropy term and the last term which takes expectations over all possible hidden state sequences under the variational posterior $q_{\mathbf{x}}(\mathbf{x}_{0:T})$. Fortunately, straight after the VBE step, we know the form of $q_{\mathbf{x}}(\mathbf{x}_{0:T})$ from (5.69), and on substituting this into $\mathrm{H}(q_{\mathbf{x}}(\mathbf{x}_{0:T}))$ we obtain

$$H(q_{\mathbf{x}}(\mathbf{x}_{0:T})) \equiv -\int d\mathbf{x}_{0:T}\; q_{\mathbf{x}}(\mathbf{x}_{0:T}) \ln q_{\mathbf{x}}(\mathbf{x}_{0:T}) \tag{5.144}$$

$$\begin{aligned}
= -\int d\mathbf{x}_{0:T}\; q_{\mathbf{x}}(\mathbf{x}_{0:T}) \Big[&- \ln Z' \\
&+ \langle \ln p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \,|\, A, B, C, D, \boldsymbol{\rho}, \boldsymbol{\mu}_0, \Sigma_0) \rangle_{q_{\boldsymbol{\theta}}(A,B,C,D,\boldsymbol{\rho})} \Big]
\end{aligned} \tag{5.145}$$

$$= \ln Z' - \langle \ln p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \,|\, A, B, C, D, \boldsymbol{\rho}, \boldsymbol{\mu}_0, \Sigma_0) \rangle_{q_{\boldsymbol{\theta}}(A,B,C,D,\boldsymbol{\rho}) q_{\mathbf{x}}(\mathbf{x}_{0:T})} \tag{5.146}$$

where the last line follows since $\ln Z'$ is not a function of the state sequence $\mathbf{x}_{0:T}$. Substituting this form (5.146) into the above form for $\mathcal{F}$ (5.142) cancels the expected complete-data term in both equations and yields a simple expression for the lower bound

$$\begin{aligned}\mathcal{F} = &-\mathrm{KL}(B) - \langle \mathrm{KL}(A \,|\, B) \rangle_{q(B)} \\ &- \mathrm{KL}(\boldsymbol{\rho}) - \langle \mathrm{KL}(D \,|\, \boldsymbol{\rho}) \rangle_{q(\boldsymbol{\rho})} - \langle \mathrm{KL}(C \,|\, \boldsymbol{\rho}, D) \rangle_{q(\boldsymbol{\rho},D)} \\ &+ \ln Z' \,. \end{aligned} \tag{5.147}$$

Note that this simpler expression is only valid straight after the VBE step. The various KL divergence terms are straightforward, yet laborious, to compute (see section C.3 for details).

We still have to evaluate the log partition function, $\ln Z'$. It is not as complicated as the integral in equation (5.70) suggests — at least in the point-parameter scenario we showed that $\ln Z' = \sum_{t=1}^{T} \ln \zeta_t(\mathbf{y}_t)$, as given in (5.83). With some care we can derive the equivalent terms $\{\zeta'_t(\mathbf{y}_t)\}_{t=1}^{T}$ for the variational Bayesian treatment, and these are given in part (c) of algorithm 5.1. Note that certain terms cancel across time steps and so the overall computation can be made more efficient if need be.

Alternatively we can calculate $\ln Z'$ from direct integration of the joint (5.70) with respect to each hidden variable one by one. In principal the hidden variables can be integrated out in any order, but at the expense of having to store statistics for many intermediate distributions.

The complete learning algorithm for state-space models is presented in algorithm 5.3. It consists of repeated iterations of the VBM step, VBE step, calculation of $\mathcal{F}$, and hyperparameter updates. In practice one does not need to compute $\mathcal{F}$ at all for learning. It may also be inefficient to update the hyperparameters after every iteration of VBEM, and for some applications in which the user is certain of their prior specifications, then a hyperparameter learning scheme may not be required at all.

5.3.8 Modifications when learning from multiple sequences

So far in this chapter the variational Bayesian algorithm has concentrated on just a data set consisting of a single sequence. For a data set consisting of n i.i.d. sequences with lengths $\{T_1, \ldots, T_n\}$, denoted $\mathbf{y} = \{\mathbf{y}_{1,1:T_1}, \ldots, \mathbf{y}_{n,1:T_n}\}$, it is straightforward to show that the VB algorithm need only be slightly modified to take into account the following changes.

Algorithm 5.3: Pseudocode for variational Bayesian state-space models.

1. Initialisation

   $\boldsymbol{\Theta} \equiv \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}\} \leftarrow$ initialise precision hyperparameters

   $\boldsymbol{\mu}_0, \Sigma_0 \leftarrow$ initialise hidden state priors

   $hss \leftarrow$ initialise hidden state sufficient statistics

2. Variational M step (VBM)

   Infer parameter posteriors $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ using $\{hss, \mathbf{y}_{1:T}, \mathbf{u}_{1:T}, \boldsymbol{\Theta}\}$

   $q(B)$, $q(A \,|\, B)$, $q(\boldsymbol{\rho})$, $q(D \,|\, \boldsymbol{\rho})$, and $q(C \,|\, \boldsymbol{\rho}, D)$

   $\overline{\boldsymbol{\phi}} \leftarrow$ calculate expected natural parameters using equations (5.52-5.67)

3. Variational E step (VBE)

   Infer distribution over hidden state $q_{\mathbf{x}}(\mathbf{x}_{0:T})$ using $\{\overline{\boldsymbol{\phi}}, \mathbf{y}_{1:T}, \mathbf{u}_{1:T}\}$

   compute $\alpha_t(\mathbf{x}_t) \equiv p(\mathbf{x}_t \,|\, \mathbf{y}_{1:t}) \quad t \in \{1, \ldots, T\}$ (forward pass, algorithm 5.1),

   compute $\beta_t(\mathbf{x}_t) \equiv p(\mathbf{y}_{t+1:T} \,|\, \mathbf{x}_t) \quad t \in \{0, \ldots, T-1\}$ (backward pass, algorithm 5.2),

   compute $\boldsymbol{\omega}_t, \Upsilon_{t,t} \quad t \in \{0, \ldots, T\}$ (marginals), and

   compute $\Upsilon_{t,t+1} \quad t \in \{0, \ldots, T-1\}$ (cross-covariance).

   $hss \leftarrow$ calculate hidden state sufficient statistics using equations (5.126-5.132)

4. Compute $\mathcal{F}$

   Compute various parameter KL divergences (appendix C.3)

   Compute log partition function, $\ln Z'$ (equation (5.70), algorithm 5.1)

   $\mathcal{F} = -\mathrm{KL}(B) - \langle \mathrm{KL}(A \,|\, B) \rangle - \mathrm{KL}(\boldsymbol{\rho}) - \langle \mathrm{KL}(D \,|\, \boldsymbol{\rho}) \rangle - \langle \mathrm{KL}(C \,|\, \boldsymbol{\rho}, D) \rangle + \ln Z'$

5. Update hyperparameters

   $\boldsymbol{\Theta} \leftarrow$ update precision hyperparameters using equations (5.133-5.136)

   $\{\boldsymbol{\mu}_0, \Sigma_0\} \leftarrow$ update auxiliary hidden state $\mathbf{x}_0$ prior hyperparameters using (5.137)

   $\{a, b\} \leftarrow$ update noise hyperparameters using (5.138)

6. While $\mathcal{F}$ is increasing, go to step 2

In the VBE step, the forward and backward passes of algorithms 5.1 and 5.2 are carried out on each sequence, resulting in a set of sufficient statistics for each of the n hidden state sequences. These are then pooled to form a combined statistic. For example, equation (5.126) becomes

$$W_A^{(i)} = \sum_{t=1}^{T_i} \langle \mathbf{x}_{i,t-1} \mathbf{x}_{i,t-1}^\top \rangle = \sum_{t=1}^{T_i} \Upsilon_{i,t-1,t-1} + \boldsymbol{\omega}_{i,t-1} \boldsymbol{\omega}_{i,t-1}^\top \,, \tag{5.148}$$

$$\text{and then} \qquad W_A = \sum_{i=1}^{n} W_A^{(i)} \,, \tag{5.149}$$

where $\Upsilon_{i,t,t}$ and $\boldsymbol{\omega}_{i,t}$ are the results of the VBE step on the ith sequence. Each of the required sufficient statistics in equations (5.126-5.132) are obtained in a similar fashion. In addition, the number of time steps T is replaced with the total over all sequences $T = \sum_{i=1}^{n} T_i$.

Algorithmically, the VBM step remains unchanged, as do the updates for the hyperparameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, a, b\}$. The updates for the hyperparameters $\boldsymbol{\mu}_0$ and Σ_0, which govern the mean and covariance of the auxiliary hidden state at time $t = 0$ for every sequence, have to be modified slightly and become

$$\boldsymbol{\mu}_0 \leftarrow \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\omega}_{i,0} \,, \tag{5.150}$$

$$\Sigma_0 \leftarrow \frac{1}{n} \sum_{i=1}^{n} \left[\Upsilon_{i,0,0} + (\boldsymbol{\mu}_0 - \boldsymbol{\omega}_{i,0})(\boldsymbol{\mu}_0 - \boldsymbol{\omega}_{i,0})^\top \right] \,, \tag{5.151}$$

where the $\boldsymbol{\mu}_0$ appearing in the update for Σ_0 is the updated hyperparameter. In the case of $n = 1$, equations (5.150) and (5.151) resemble their originals forms given in section 5.3.6. Note that these batch updates trivially extend the analogous result for ML parameter estimation of linear dynamical systems presented by Ghahramani and Hinton (Ghahramani and Hinton, 1996a, equation (25)), since here we do not assume that the sequences are equal in length (it is clear from the forward and backward algorithms in both the ML and VB implementations that the posterior variance of the auxiliary state $\Upsilon_{i,0,0}$ will only be constant if all the sequences have the same length).

Finally the computation of the lower bound $\mathcal{F}$ is unchanged except that it now involves a contribution from each sequence

$$\begin{aligned} \mathcal{F} = &- \mathrm{KL}(B) - \langle \mathrm{KL}(A \,|\, B) \rangle_{q(B)} \\ &- \mathrm{KL}(\boldsymbol{\rho}) - \langle \mathrm{KL}(D \,|\, \boldsymbol{\rho}) \rangle_{q(\boldsymbol{\rho})} - \langle \mathrm{KL}(C \,|\, \boldsymbol{\rho}, D) \rangle_{q(\boldsymbol{\rho}, D)} + \sum_{i=1}^{n} \ln Z'^{(i)} \,, \end{aligned}$$

where $\ln Z'^{(i)}$ is computed in the VBE step in algorithm 5.1 for each sequence individually.

5.3.9 Modifications for a fully hierarchical model

As mentioned towards the end of section 5.2.2, the hierarchy of hyperparameters for priors over the parameters is not complete for this model as it stands. There remains the undesirable feature that the parameters Σ_0 and $\boldsymbol{\mu}_0$ contain more free parameters as the dimensionality of the hidden state increases. There is a similar problem for the precision hyperparameters. We refer the reader to chapter 4 in which a similar structure was used for the hyperparameters of the factor loading matrices.

With such variational distributions in place for VB LDS, the propagation algorithms would change, replacing, for example, $\boldsymbol{\alpha}$, with its expectation over its variational posterior, $\langle \boldsymbol{\alpha} \rangle_{q(\boldsymbol{\alpha})}$, and the hyperhyperparameters a_α, b_α of equation (5.17) would be updated to best fit the variational posterior for α, in the same fashion that the hyperparameters a, b are updated to reflect the variational posterior on ρ (section 5.3.6). In addition a similar KL penalty term would arise.

For the parameters Σ_0 and $\boldsymbol{\mu}_0$, again KL terms would crop up in the lower bound, and where these quantities appeared in the propagation algorithms they would have to be replaced with their expectations under their variational posterior distributions.

These modifications were considered too time-consuming to implement for the experiments carried out in the following section, and so we should of course be mindful of their exclusion.

5.4 Synthetic Experiments

In this section we give two examples of how the VB algorithm for linear dynamical systems can discover meaningful structure from the data. The first example is carried out on a data set generated from a simple LDS with no inputs and a small number of hidden states. The second example is more challenging and attempts to learn the number of hidden states and their dynamics in the presence of noisy inputs. We find in both experiments that the ARD mechanism which optimises the precision hyperparameters can be used successfully to determine the structure of the true generating model.

5.4.1 Hidden state space dimensionality determination (no inputs)

An LDS with hidden state dimensionality of $k = 6$ and an output dimensionality of $p = 10$ was set up with parameters randomly initialised according to the following procedure.

The dynamics matrix A $(k \times k)$ was fixed to have eigenvalues of $(.65, .7, .75, .8, .85, .9)$, constructed from a randomly rotated diagonal matrix; choosing fairly high eigenvalues ensures that

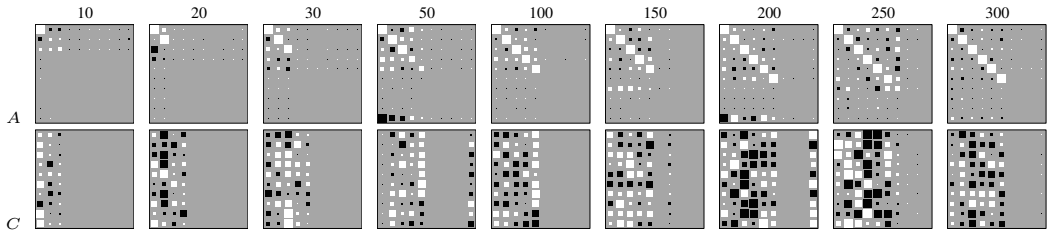
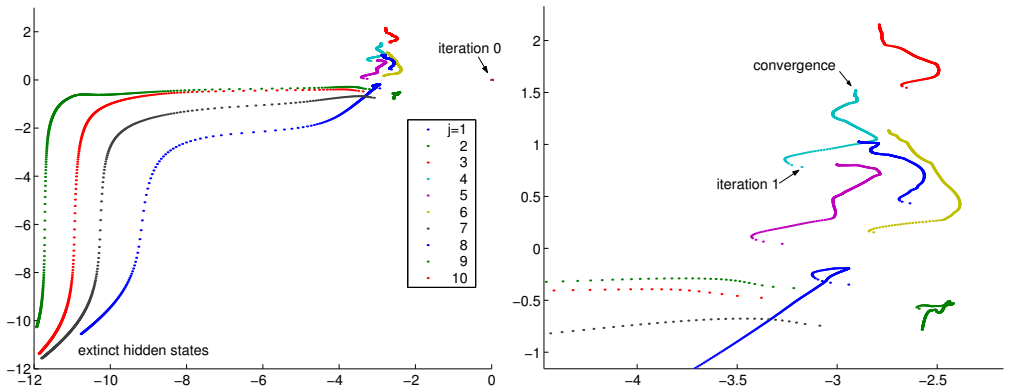


Figure 5.4: Hinton diagrams of the dynamics (A) and output (C) matrices after 500 iterations of VBEM. From left to right, the length of the observed sequence $\mathbf{y}_{1:T}$ increases from $T = 10$ to 300. This true data was generated from a linear dynamical system with $k = 6$ hidden state dimensions, all of which participated in the dynamics (see text for a description of the parameters used). As a visual aid, the entries of A matrix and the columns of the C matrix have been permuted in the order of the size of the hyperparameters in $\boldsymbol{\gamma}$.

every dimension participates in the hidden state dynamics. The output matrix C $(p \times k)$ had each entry sampled from a bimodal distribution made from a mixture of two Gaussians with means at (2,-2) and common standard deviations of 1; this was done in an attempt to keep the matrix entries away from zero, such that every hidden dimension contributes to the output covariance structure. Both the state noise covariance Q and output noise covariance R were set to be the identity matrix. The hidden state at time $t = 1$ was sampled from a Gaussian with mean zero and unit covariance.

From this LDS model several training sequences of increasing length were generated, ranging from $T = 10, \ldots, 300$ (the data sets are incremental). A VBLDS model with hidden state space dimensionality $k = 10$ was then trained on each single sequence, for a total of 500 iterations of VBEM. The resulting A and C matrices are shown in figure 5.4. We can see that for short sequences the model chooses a simple representation of the dynamics and output processes, and for longer sequences the recovered model is the same as the underlying LDS model which generated the sequences. Note that the model learns a predominantly diagonal dynamics matrix, or a self-reinforcing dynamics (this is made obvious by the permutation of the states in the figure (see caption), but is not a contrived observation). The likely reason for this is the prior's preference for the A matrix to have small sum-of-square entries for each column; since the dynamics matrix has to capture a certain amount of power in the hidden dynamics, the least expensive way to do this is to place most of the power on the diagonal entries.

Plotted in figure 5.5 are the trajectories of the hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, during the VB optimisation for the sequence of length $T = 300$. For each hidden dimension j the output hyperparameter γ_j (vertical) is plotted against the dynamics hyperparameter α_j. It is in fact the logarithm of the *reciprocal* of the hyperparameter that is plotted on each axis. Thus if a hidden dimension becomes extinct, the reciprocal of its hyperparameter tends to zero (bottom left of plots). Each component of each hyperparameter is initialised to 1 (see annotation for iteration 0, at top right of plot 5.5(a)), and during the optimisation some dimensions become extinct. In this example, four hidden state dimensions become extinct, both in their ability to participate in the dynamics



(a) Hidden state inverse-hyperparameter trajectories (logarithmic axes).

(b) Close-up of top right corner of (a).

Figure 5.5: Trajectories of the hyperparameters for the case $n = 300$, plotted as $\ln \frac{1}{\alpha}$ (horizontal axis) against $\ln \frac{1}{\gamma}$ (vertical axis). Each trace corresponds to one of k hidden state dimensions, with points plotted after each iteration of VBEM. Note the initialisation of $(1, 1)$ for all $(\alpha_j, \gamma_j),\ j = 1, \ldots, k$ (labelled iteration 0). The direction of each trajectory can be determined by noting the spread of positions at successive iterations, which are resolvable at the beginning of the optimisation, but not so towards the end (see annotated close-up). Note especially that four hyperparameters are flung to locations corresponding to very small variances of the prior for both the A and C matrix columns (i.e. this has effectively removed those hidden state dimensions), and six remain in the top right with finite variances. Furthermore, the L-shaped trajectories of the eventually extinct hidden dimensions imply that in this example the dimensions are removed first from the model's dynamics, and then from the output process (see figure 5.8(a,c) also).

and their contribution to the covariance of the output data. Six hyperparameters remain useful, corresponding to $k = 6$ in the true model. The trajectories of these are seen more clearly in figure 5.5(b).

5.4.2 Hidden state space dimensionality determination (input-driven)

This experiment demonstrates the capacity of the input-driven model to use (or not to use) an input-sequence to model the observed data. We obtained a sequence $\mathbf{y}_{1:T}$ of length $T = 100$ by running the linear dynamical system as given in equations (5.4,5.5), with a hidden state space dimensionality of $k = 2$, generating an observed sequence of dimensionality $p = 4$. The input sequence, $\mathbf{u}_{1:T}$, consisted of three signals: the first two were $\frac{\pi}{2}$ phase-lagged sinusoids of period 50, and the third dimension was uniform noise $\sim \mathrm{U}(0, 1)$.

The parameters A, C, and R were created as described above (section 5.4.1). The eigenvalues of the dynamics matrix were set to $(.65, .7)$, and the covariance of the hidden state noise set to the identity. The parameter B $(k \times u)$ was set to the all zeros matrix, so the inputs did not modulate

the hidden state dynamics. The first two columns of the D $(p \times u)$ matrix were sampled from the uniform $\mathrm{U}(-10, 10)$, so as to induce a random (but fixed) displacement of the observation sequence. The third column of the D matrix was set to zeros, so as to ignore the third input dimension (noise). Therefore the only noise in the training data was that from the state and output noise mechanisms (Q and R).

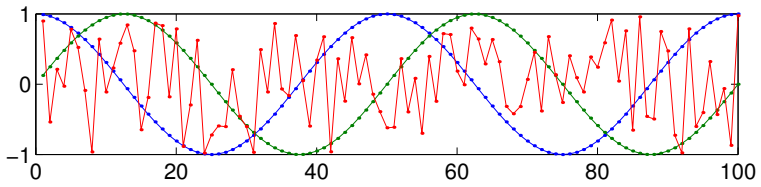
Figure 5.6 shows the input sequence used, the generated hidden state sequence, and the resulting observed data, over $T = 100$ time steps. We would like the variational Bayesian linear dynamical system to be able to identify the number of hidden dimensions required to model the observed data, taking into account the modulatory effect of the input sequence. As in the previous experiment, in this example we attempt to learn an over-specified model, and make use of the ARD mechanisms in place to recover the structure of the underlying model that generated the data.

In full, we would like the model to learn that there are $k = 2$ hidden states, that the third input dimension is irrelevant to predicting the observed data, that all the input dimensions are irrelevant for the hidden state dynamics, and that it is only the two dynamical hidden variables that are being embedded in the data space.

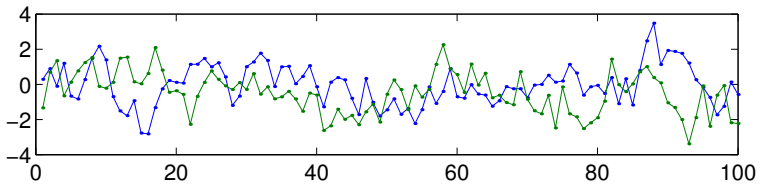
The variational Bayesian linear dynamical system was run with $k = 4$ hidden dimensions, for a total of 800 iterations of VBE and VBM steps (see algorithm 5.3 and its sub-algorithms). Hyperparameter optimisations after each VBM step were introduced on a staggered basis to ease interpretability of the results. The dynamics-related hyperparameter optimisations (i.e. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) were begun after the first 10 iterations, the output-related optimisations (i.e. $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$) after 20 iterations, and the remaining hyperparameters (i.e. a, b, Σ_0 and $\boldsymbol{\mu}_0$) optimised after 30 iterations. After each VBE step, $\mathcal{F}$ was computed and the current state of the hyperparameters recorded.

Figure 5.7 shows the evolution of the lower bound on the marginal likelihood during learning, displayed as both the value of $\mathcal{F}$ computed after each VBE step (figure 5.7(a)), and the *change* in $\mathcal{F}$ between successive iterations of VBEM (figure 5.7(b)). The logarithmic plot shows the onset of each group of hyperparameter optimisations (see caption), and also clearly shows three regions where parameters are being pruned from the model.

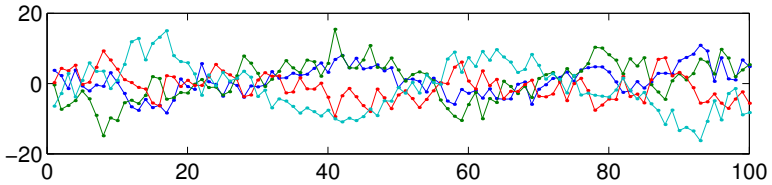
As before we can analyse the change in the hyperparameters during the optimisation process. In particular we can examine the ARD hyperparameter vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}$, which contain the prior precisions for the entries of each column of each of the matrices A, B, C and D respectively. Since the hyperparameters are updated to reflect the variational posterior distribution over the parameters, a large value suggest that the relevant column contains entries are close to zero, and therefore can be considered excluded from the state-space model equations (5.4) and (5.5).



(a) 3 dimensional input sequence.

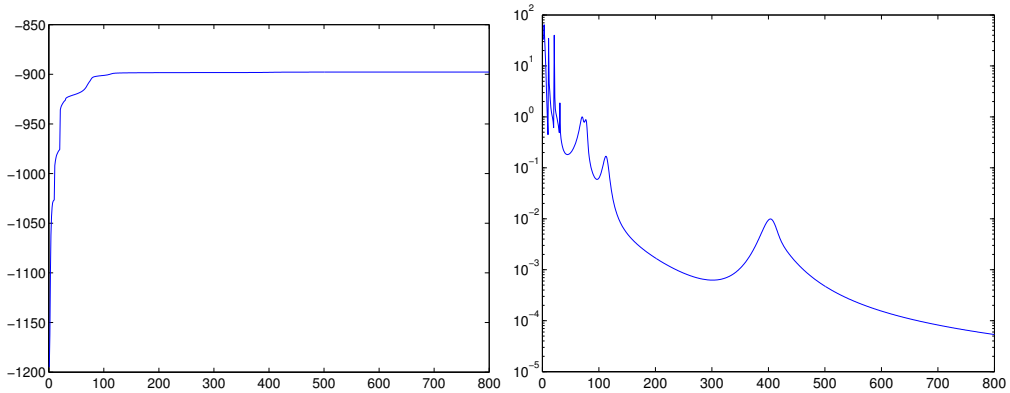


(b) 2 dimensional hidden state sequence.



(c) 4 dimensional observed data.

Figure 5.6: Data for the input-driven example in section 5.4.2. **(a)**: The 3 dimensional input data consists of two phase-lagged sinusoids of period 50, and a third dimension consisting of noise uniformly distributed on $[0, 1]$. Both B and D contain zeros in their third columns, so the noise dimension is not used when generating the synthetic data. **(b)**: The hidden state sequence generated from the dynamics matrix, A, which in this example evolves independently of the inputs. **(c)**: The observed data, generated by combining the embedded hidden state sequence (via the output matrix C) and the input sequence (via the input-output matrix D), and then adding noise with covariance R. Note that the observed data is now a sinusoidally modulated simple linear dynamical system.



(a) Evolution of $\mathcal{F}$ during iterations of VBEM. (b) Change in $\mathcal{F}$ between successive iterations.

Figure 5.7: Evolution of the lower bound $\mathcal{F}$ during learning of the input-dependent model of section 5.4.2. **(a)**: The lower bound $\mathcal{F}$ increases monotonically with iterations of VBEM. **(b)**: Interesting features of the optimisation can be better seen in a logarithmic plot of the change of $\mathcal{F}$ between successive iterations of VBEM. For example, it is quite clear there is a sharp increase in $\mathcal{F}$ at 10 iterations (dynamics-related hyperparameter optimisation activated), at 20 iterations (output-related hyperparameter optimisation activated), and at 30 iterations (the remaining hyperparameter optimisations are activated). The salient peaks around 80, 110, and 400 iterations each correspond to the gradual automatic removal of one or more parameters from the model by hyperparameter optimisation. For example, it is quite probable that the peak at around iteration 400 is due to the recovery of the first hidden state modelling the dynamics (see figure 5.8).

Figure 5.8 displays the components of each of the four hyperparameter vectors throughout the optimisation. The reciprocal of the hyperparameter is plotted since it is more visually intuitive to consider the variance of the parameters falling to zero as corresponding to extinction, instead of the precision growing without bound. We can see that, by 500 iterations, the algorithm has (correctly) discovered that there are only two hidden variables participating in the dynamics (from $\boldsymbol{\alpha}$), these same two variables are used as factors embedded in the output (from $\boldsymbol{\gamma}$), that none of the input dimensions is used to modulate the hidden dynamics (from $\boldsymbol{\beta}$), and that just two dimensions of the input are required to displace the data (from $\boldsymbol{\delta}$). The remaining third dimension of the input is in fact disregarded completely by the model, which is exactly according to the recipe used for generating this synthetic data.

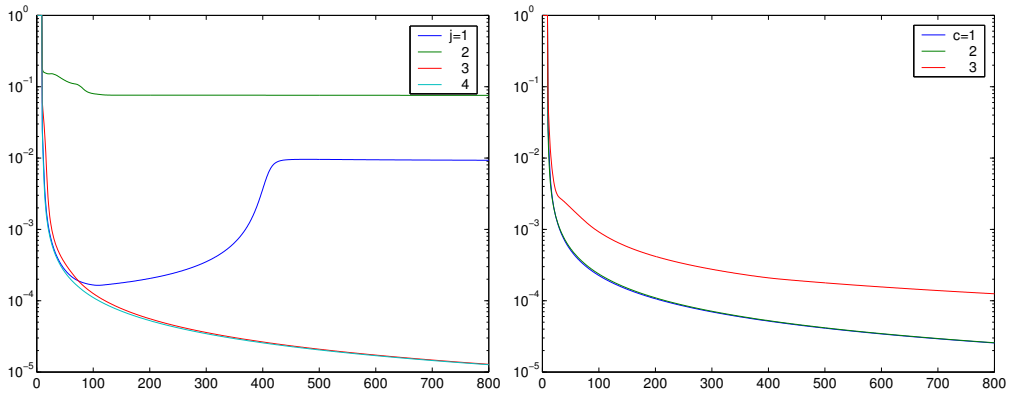
Of course, with a smaller data set, the model may begin to remove some parameters corresponding to arcs of influence between variables across time steps, or between the inputs and the dynamics or outputs. This and the previous experiment suggest that with enough data, the algorithm will generally discover a good model for the data, and indeed recover the true (or equivalent) model if the data was in fact generated from a model within the class of models accessible by the specified input-dependent linear dynamical system.

Although not observed in the experiment presented here, some caution needs to be taken with much larger sequences to avoid local minima in the optimisation. In the larger data sets the problems of local maxima or very long plateau regions in the optimisation become more frequent, with certain dimensions of the latent space modelling either the dynamics or the output processes, but not both (or neither). This problem is due to the presence of a dynamics model coupling the data across each time step. Recall that in the factor analysis model (chapter 4), because of the spherical factor noise model, ARD can rotate the factors into a basis where the outgoing weights for some factors can be set to zero (by taking their precisions to infinity). Unfortunately this degeneracy is not present for the hidden state variables of the LDS model, and so concerted efforts are required to rotate the hidden state along the entire sequence.

5.5 Elucidating gene expression mechanisms

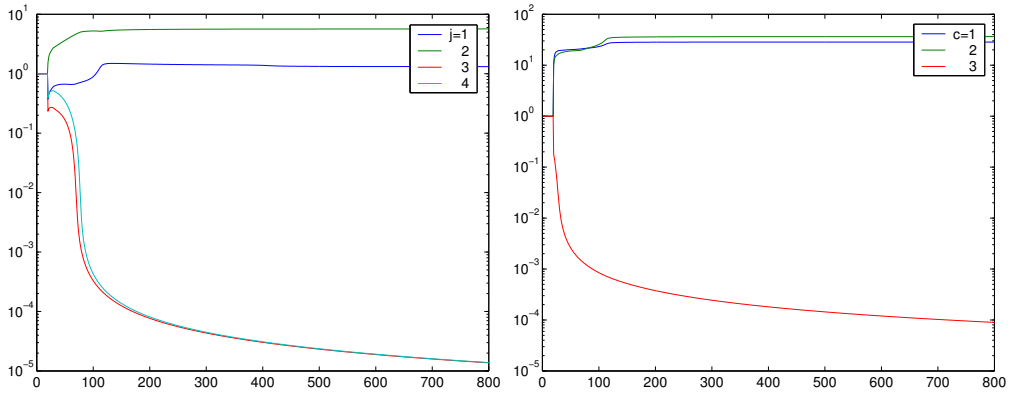
Description of the process and data

The data consists of $n = 34$ time series of the expressions of genes involved in a transcriptional process in the nuclei of human T lymphocytes. Each sequence consists of $T = 10$ measurements of the expressions of $p = 88$ genes, at time points $(0, 2, 4, 6, 8, 18, 24, 32, 48, 72)$ hours after a treatment to initiate the transcriptional process (see Rangel et al., 2001, section 2.1). For each sequence, the expression levels of each gene were normalised to have mean 1, by dividing by the mean gene expression over the 10 time steps. This normalisation reflects our interest in



(a) Prior variance on each column of A, $\frac{1}{\boldsymbol{\alpha}}$.

(b) Prior variance on each column of B, $\frac{1}{\boldsymbol{\beta}}$.



(c) Prior variance on each column of C, $\frac{1}{\boldsymbol{\gamma}}$.

(d) Prior variance on each column of D, $\frac{1}{\boldsymbol{\delta}}$.

Figure 5.8: Evolution of the hyperparameters with iterations of variational Bayesian EM, for the input-driven model trained on the data shown in figure 5.6 (see section 5.4.2). Each plot shows the reciprocal of the components of a hyperparameter vector, corresponding to the prior variance of the entries of each column of the relevant matrix. The hyperparameter optimisation is activated after 10 iterations of VBEM for the dynamics-related hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, after 20 iterations for the output-related hyperparameters $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$, and after 30 for the remaining hyperparmeters. **(a)**: After 150 iterations of VBEM, $\frac{1}{\alpha_3} \to 0$ and $\frac{1}{\alpha_4} \to 0$, which corresponds to the entries in the 3rd and 4th columns of A tending to zero. Thus only the remaining two hidden dimensions (1,2) are being used for the dynamics process. **(b)**: All hyperparameters in the $\boldsymbol{\beta}$ vector grow large, corresponding to each of the column entries in B being distributed about zero with high precision; thus none of the dimensions of the input vector is being used to modulate the hidden state. **(c)**: Similar to the A matrix, two hyperparameters in the vector $\boldsymbol{\gamma}$ remain small, and the remaining two increase without bound, $\frac{1}{\gamma_3} \to 0$ and $\frac{1}{\gamma_4} \to 0$. This corresponds to just two hidden dimensions (factors) causing the observed data through the C embedding. These are the *same* dimensions as used for the dynamics process, agreeing with the mechanism that generated the data. **(d)**: Just one hyperparameter, $\frac{1}{\delta_3} \to 0$, corresponding to the model ignoring the third dimension of the input, which is a confusing input unused in the true generation process (as can be seen from figure 5.6(a)). Thus the model learns that this dimension is irrelevant to modelling the data.

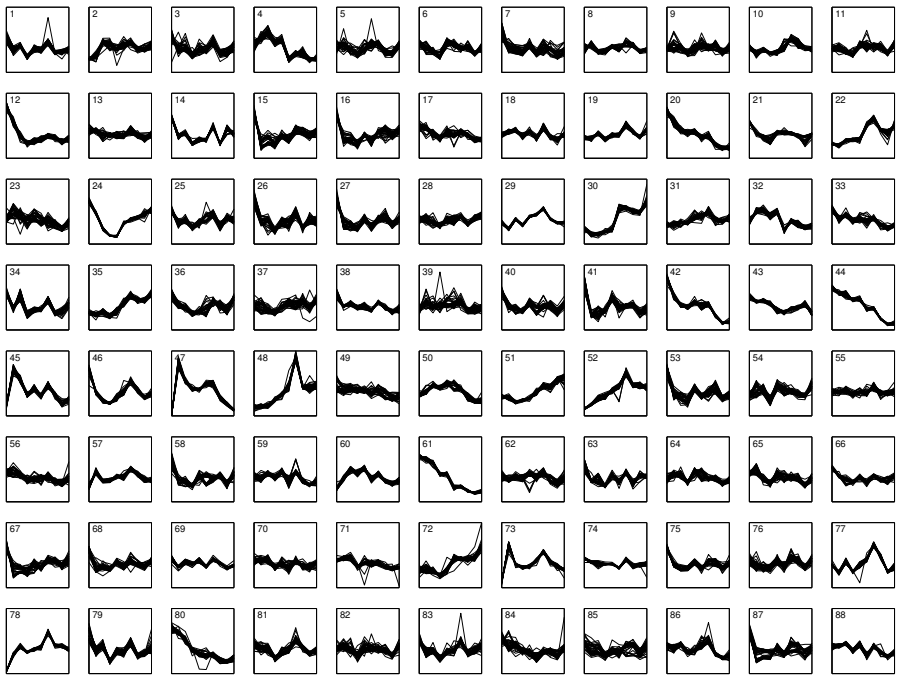


Figure 5.9: The gene expression data of Rangel et al. (2001). Each of the 88 plots corresponds to a particular gene on the array, and contains all of the recorded 34 sequences each of length 10.

the profiles of the genes rather than the absolute expression levels. Figure 5.9 shows the entire collection of normalised expression levels for each gene.

A previous approach to modelling gene expression levels which used graphical models to model the causal relationships between genes is presented in Friedman et al. (2000). However, this approach ignored the temporal dependence of the gene intensities during trials and went only as far as to infer the causal relationships between the genes within one time step. Their method discretised expression levels and made use of efficient candidate proposals and greedy methods for searching the space of model structures. This approach also assumed that all the possibly interacting variables are observed on the microarray. This precludes the existence of hidden causes or unmeasured genes whose involvement might dramatically simplify the network structure and therefore ease interpretability of the mechanisms in the underlying biological process.

Linear dynamical systems and other kinds of possibly nonlinear state-space models are a good class of model to begin modelling this gene expression data. The gene expression measurements are the noisy 88-dimensional outputs of the linear dynamical system, and the hidden states of the model correspond to unobserved factors in the gene transcriptional process which are not recorded in the DNA microarray — they might correspond simply to unmeasured genes, or they could model more abstractly the effect of players other than genes, for example regulatory proteins and background processes such as mRNA degradation.

Some aspects of using the LDS model for this data are not ideal. For example, we make the assumptions that the dynamics and output processes are time invariant, which is unlikely in a real biological system. Furthermore the times at which the data are taken are not linearly-spaced (see above), which might imply that there is some (possibly well-studied) non-linearity in the rate of the transcriptional process; worse still, there may be whole missing time slices which, if they had been included, would have made the dynamics process closer to stationary. There is also the usual limitation that the noise in the dynamics and output processes is almost certainly not Gaussian.

Experiment results

In this experiment we use the input-dependent LDS model, and *feed back* the gene expressions from the previous time step into the input for the current time step; in doing so we attempt to discover gene-gene interactions across time steps (in a causal sense), with the hidden state in this model now really representing unobserved variables. An advantage of this architecture is that we can now use the ARD mechanisms to determine which genes are influential across adjacent time slices, just as before (in section 5.4.2) we determined which inputs were relevant to predicting the data.

A graphical model for this setup is given in figure 5.10. When the input is replaced with the previous time step's observed data, the equations for the state-space model can be rewritten from equations (5.4) and (5.5) into the form:

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{y}_{t-1} + \mathbf{w}_t \tag{5.152}$$

$$\mathbf{y}_t = C\mathbf{x}_t + D\mathbf{y}_{t-1} + \mathbf{v}_t \ . \tag{5.153}$$

As a function only of the data at the previous time step, $\mathbf{y}_{t-1}$, the data at time t can be written

$$\mathbf{y}_t = (CB + D)\mathbf{y}_{t-1} + \mathbf{r}_t \ , \tag{5.154}$$

where $\mathbf{r}_t = \mathbf{v}_t + C\mathbf{w}_t + CA\mathbf{x}_{t-1}$ includes all contributions from noise and previous states. Thus to first order the interaction between gene d and gene a can be characterised by the element $[CB + D]_{ad}$ of the matrix. Indeed this matrix need not be symmetric and the element represents activation or inhibition from gene d to gene a at the next time step, depending on its sign. We will return to this quantity shortly.

5.5.1 Generalisation errors

For this experiment we trained both variational Bayesian and MAP LDS models on the first 30 of the 34 gene sequences, with the dimension of the hidden state ranging from $k = 1$ to

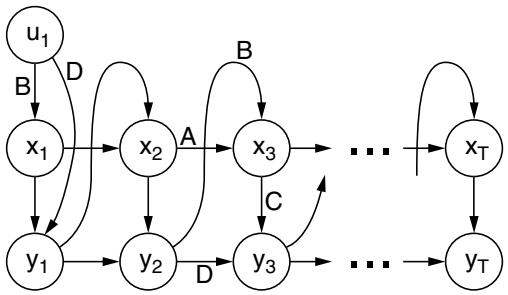


Figure 5.10: The feedback graphical model with outputs feeding into inputs.

20. The remaining 4 sequences were set aside as a test set. Since we required an input at time $t = 1$, $\mathbf{u}_1$, the observed sequences that were learnt began from time step $t = 2$. The MAP LDS model was implemented using the VB LDS with the following two modifications: first, the hyperparameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}$ and a, b were not optimised (however, the auxiliary state prior mean $\boldsymbol{\mu}_0$ and covariance Σ_0 were learnt); second, the sufficient statistics for the parameters were artificially boosted by a large factor to simulate delta functions for the posterior — i.e. in the limit of large n the VBM step recovers the MAP M step estimate of the parameters.

Both algorithms were run for 300 EM iterations, with no restarts. The one-step-ahead mean total square reconstruction error was then calculated for both the training sequences and the test sequences using the learnt models; the reconstruction of the tth observation for the ith sequence, $\mathbf{y}_{i,t}$, was made like so:

$$\hat{\mathbf{y}}_{i,t}^{\text{MAP}} = C_{\text{MAP}} \langle \mathbf{x}_{i,t} \rangle_{q_{\mathbf{x}}} + D_{\text{MAP}} \mathbf{y}_{i,t-1} \tag{5.155}$$

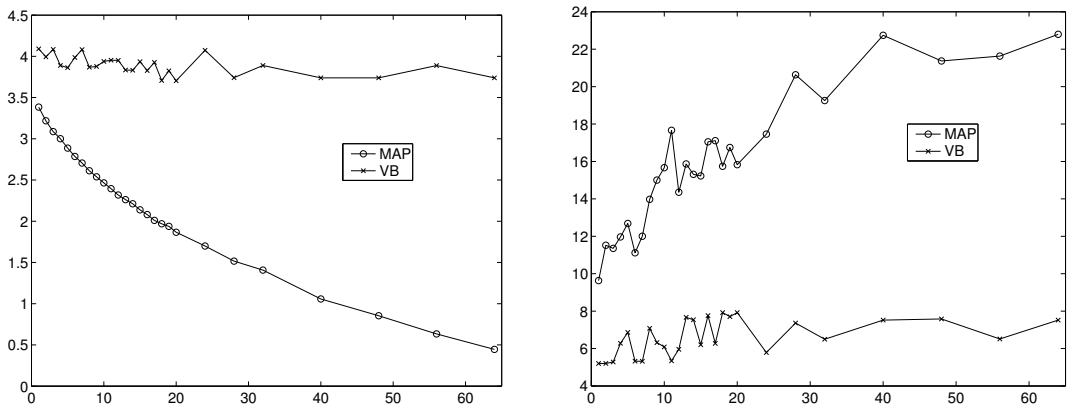
$$\hat{\mathbf{y}}_{i,t}^{\text{VB}} = \langle C \rangle_{q_C} \langle \mathbf{x}_{i,t} \rangle_{q_{\mathbf{x}}} + \langle D \rangle_{q_D} \mathbf{y}_{i,t-1} \ . \tag{5.156}$$

To clarify the procedure: to reconstruct the observations for the ith sequence, we use the entire observation sequence $\mathbf{y}_{i,1:T}$ to first infer the distribution over the hidden state sequence $\mathbf{x}_{i,1:T}$, and then we attempt to reconstruct each $\mathbf{y}_{i,t}$ using just the hidden state $\mathbf{x}_{i,t}$ and $\mathbf{y}_{i,t-1}$. The form given for the VB reconstruction in (5.156) is valid since, subject to the approximate posterior: all of the variational posterior distributions over the parameters and hidden states are Gaussian, C and $\mathbf{x}_t$ are independent, and the noise is Student-t distributed with mean zero.

Thus for each value of k, and for each of the MAP and VB learnt models, the total squared error per sequence is calculated according to:

$$E_{\text{train}} = \frac{1}{n_{\text{train}}} \sum_{i \in \text{train}} \sum_{t=2}^{T_i} (\hat{\mathbf{y}}_{i,t} - \mathbf{y}_{i,t})^2 \tag{5.157}$$

$$E_{\text{test}} = \frac{1}{n_{\text{test}}} \sum_{i \in \text{test}} \sum_{t=2}^{T_i} (\hat{\mathbf{y}}_{i,t} - \mathbf{y}_{i,t})^2 \ . \tag{5.158}$$



(a) Training set error per sequence.

(b) Test set error per sequence.

Figure 5.11: The per sequence squared reconstruction error for one-step-ahead prediction (see text), as a function of the dimension of the hidden state, ranging from $k = 1$ to 64, on **(a)** the 30 training sequences, and **(b)** the 4 test sequences.

Figure 5.11 shows the squared reconstruction error for one-step-ahead prediction, as a function of the dimension of the hidden state for both the training and test sequences. We see that the MAP LDS model achieves a decreasing reconstruction error on the training set as the dimensionality of the hidden state is increased, whereas VB produces an approximately constant error, albeit higher. On prediction for the test set, MAP LDS performs very badly and increasingly worse for more complex learnt models, as we would expect; however, the VB performance is roughly constant with increasing k, suggesting that VB is using the ARD mechanism successfully to discard surplus modelling power. The test squared prediction error is slightly more than that on the training set, suggesting that VB is overfitting slightly.

5.5.2 Recovering gene-gene interactions

We now return to the interactions between genes d and a – more specifically the influence of gene d on gene a – in the matrix $[CB + D]$. Those entries in the matrix which are significantly different from zero can be considered as candidates for 'interactions'. Here we consider an entry to be significant if the zero point is more than 3 standard deviations from the posterior mean for that entry (based on the variational posterior distribution for the entry). Calculating the significance for the combined $CB+D$ matrix is laborious, and so here we provide results for only the D matrix. Since there is a degeneracy in the feedback model, we chose to effectively remove the first term, CB, by constraining all (but one) of the hyperparameters in $\boldsymbol{\beta}$ to very high values. The spared hyperparameter in $\boldsymbol{\beta}$ is used to still model an offset in the hidden dynamics using the bias input. This process essentially enforces $[CB]_{ad} = 0$ for all gene-gene pairs, and so simplifies the interpretation of the learnt model.

Figure 5.12 shows the interaction matrix learnt by the MAP and VB models (with the column corresponding the bias removed), for the case of $k = 2$ hidden state dimensions. For the MAP result we simply show $D + CB$. We see that the MAP and VB matrices share some aspects in terms of the signs and size of some of the interactions, but under the variational posterior only a few of the interactions are significantly non-zero, leading to a very sparse interaction matrix (see figure 5.13). Unfortunately, due to proprietary restrictions on the expression data the identities of the genes cannot be published here, so it is hard to give a biological interpretation to the network in figure 5.13. The hope is that these graphs suggest interactions which agree qualitatively with the transcriptional mechanisms already established in the research community. The ultimate result would be to be able to confidently predict the existence of as-yet-undocumented mechanisms to stimulate and guide future biological experiments. The VB LDS algorithm may provide a useful starting point for this research programme.

5.6 Possible extensions and future research

The work in this chapter can be easily extended to linear-Gaussian state-space models on trees, rather than chains, which could be used to model a variety of data. Moreover, for multiply-connected graphs, the VB propagation subroutine can still be used within a structured VB approximation.

Another interesting application of this body of theory could be to a Bayesian version of what we call a *switching state-space model* (SwSSM), which has the following dynamics:

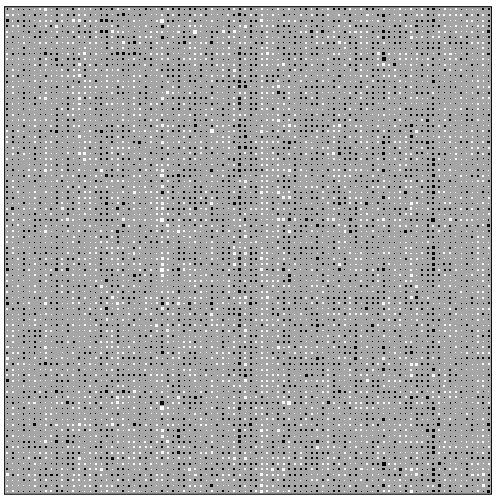
$$\text{a switch variable } s_t \text{ with dynamics} \quad p(s_t = i \,|\, s_{t-1} = j) = T_{ij} \;, \tag{5.159}$$

$$\text{hidden state dynamics} \quad p(\mathbf{x}_t \,|\, s_{t-1}, \mathbf{x}_{t-1}) = \mathrm{N}(\mathbf{x}_t \,|\, A_{s_{t-1}}\mathbf{x}_{t-1}, Q_{s_{t-1}}) \;, \tag{5.160}$$

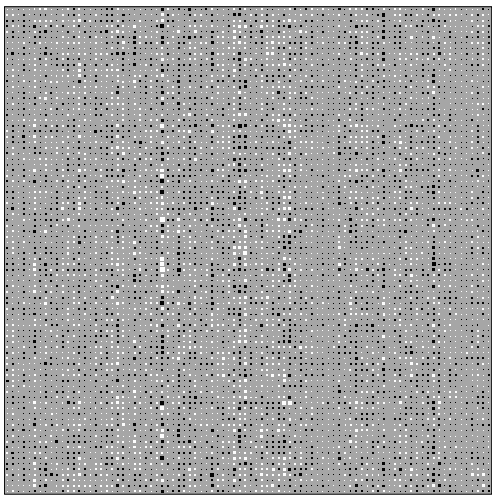
$$\text{and output function} \quad p(\mathbf{y}_t \,|\, s_t, \mathbf{x}_t) = \mathrm{N}(\mathbf{y}_t \,|\; C_{s_t}\mathbf{x}_t, R_{s_t}) \;. \tag{5.161}$$

That is to say we have a non-stationary switching linear dynamical system whose parameters are drawn from a finite set according to a discrete variable with its own dynamics. The appealing aspect of this model is that it contains many models as special cases, including: mixtures of factor analysers, mixtures of linear dynamical systems, Gaussian-output hidden Markov models, and mixtures of Gaussians. With appropriate optimisation of the lower bound on the marginal likelihood, one would hope that the data would provide evidence that one or other, or indeed hybrids, of the above special cases was the underlying generating model, or best approximates the true generating process in some sense. We have seen an example of variational Bayesian learning for hidden Markov models in chapter 3.

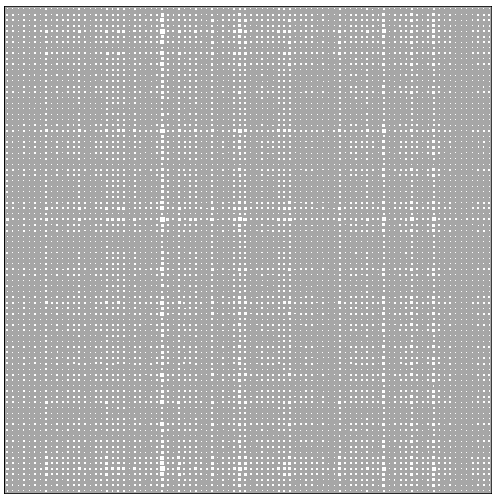
We have not commented on how reliably we expect the variational Bayesian method to approximate the marginal likelihood. Indeed a full analysis of the tightness of the variational bound



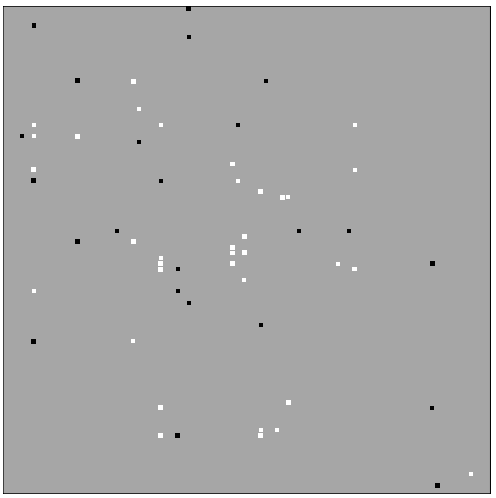
(a) The MAP EM solution $[D + CB]_{ad}$.



(b) Means $\langle D_{ad} \rangle$ after VBEM.



(c) Variances $\langle D_{ad}^2 \rangle - \langle D_{ad} \rangle^2$ after VBEM.



(d) Significant entries of D under $q_D(D)$.

Figure 5.12: The gene-gene interaction matrix learnt from the **(a)** MAP and **(b)** VB models (with the column corresponding to the bias input removed). Note that some of the entries are similar in each of the two matrices. Also shown is **(c)** the covariance of the posterior distribution of each element, which is a separable product of functions of each of the two genes' identities. Show in **(d)** are the entries of $\langle D_{ad} \rangle$ which are significantly far from zero, that is the value of zero is more than 3 standard deviations from the mean of the posterior.

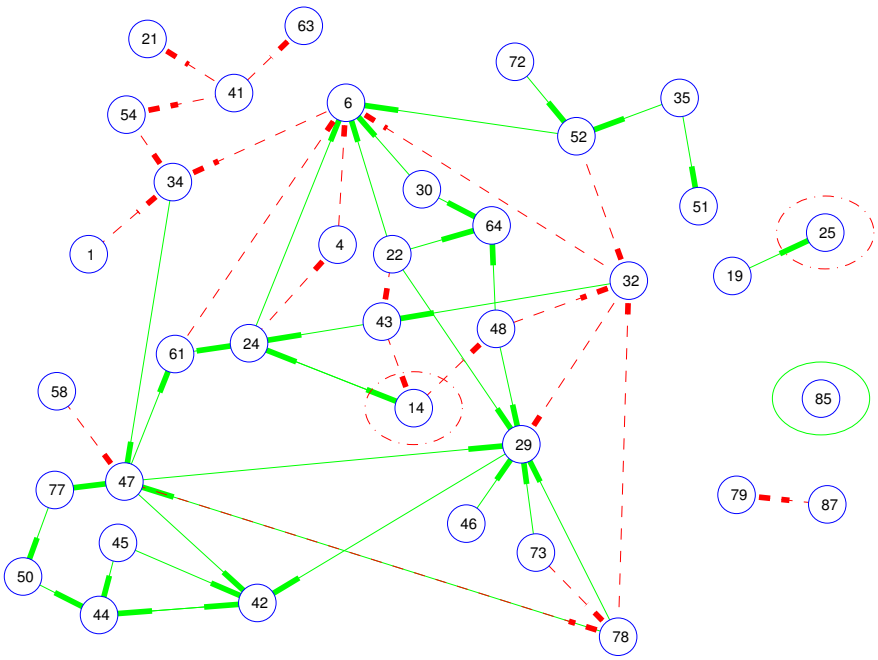


Figure 5.13: An example representation of the recovered interactions in the D matrix, as shown in figure 5.12(d). Each arc between two genes represents a significant entry in D. Red (dotted) and green (solid) denote inhibitory and excitatory influences, respectively. The direction of the influence is from the the thick end of the arc to the thin end. Ellipses denote self-connections. To generate this plot the genes were placed randomly and then manipulated slightly to reduce arc-crossing.

would require sampling for this model (as carried out in Früwirth-Schnatter, 1995, for example). This is left for further work, but the reader is referred to chapter 4 of this thesis and also to Miskin (2000), where sampling estimates of the marginal likelihood are directly compared to the VB lower bound and found to be comparable for practical problems.

We can also model higher than first order Markov processes using this model, by extending the feedback mechanism used in section 5.5. This could be achieved by feeding back concatenated observed data $\mathbf{y}_{t-d:t-1}$ into the current input vector u_t, where d is related to the maximum order present in the data. This procedure is common practice to model higher order data, but in our Bayesian scheme we can also learn posterior uncertainties for the parameters of the feedback, and can entirely remove some of the inputs via the hyperparameter optimisation.

This chapter has dealt solely with the case of linear dynamics and linear output processes with Gaussian noise. Whilst this is a good first approximation, there are many scenarios in which a non-linear model is more appropriate, for one or both of the processes. For example, Särelä et al. (2001) present a model with factor analysis as the output process and a two layer MLP network to model a non-linear dynamics process from one time step to the next, and Valpola and Karhunen (2002) extend this to include a non-linear output process as well. In both, the posterior is assumed to be of (constrained) Gaussian form and a variational optimisation is performed to learn the parameters and infer the hidden factor sequences. However, their model does not exploit the full forward-backward propagation and instead updates the hidden state one step forward and backward in time at each iteration.

5.7 Summary

In this chapter we have shown how to approximate the marginal likelihood of a Bayesian linear dynamical system using variational methods. Since the complete-data likelihood for the LDS model is in the conjugate-exponential family it is possible to write down a VBEM algorithm for inferring the hidden state sequences whilst simultaneously maintaining uncertainty over the parameters of the model, subject to the approximation that the hidden variables and parameters are independent given the data.

Here we have had to rederive the forward and backward passes in the VBE step in order for them to take as input the natural parameter expectations from the VBM step. It is an open problem to prove that for LDS models the natural parameter mapping $\boldsymbol{\phi}(\boldsymbol{\theta})$ is not invertible; that is to say there exists no $\tilde{\boldsymbol{\theta}}$ in general that satisfies $\boldsymbol{\phi}(\tilde{\boldsymbol{\theta}}) = \overline{\boldsymbol{\phi}} = \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{q_{\boldsymbol{\theta}}(\boldsymbol{\theta})}$. We have therefore derived here the variational Bayesian counterparts of the Kalman filter and Rauch-Tung-Striebel smoother, which can in fact be supplied with *any* distribution over the parameters. As with other conjugate-exponential VB treatments, the propagation algorithms have the same complexity as the MAP point-parameter versions.

We have shown how the algorithm can use the ARD procedure of optimising precision hyperparameters to discover the structure of models of synthetic data, in terms of the number of required hidden dimensions. By feeding back previous data into the inputs of the model we have shown how it is possible to elucidate interactions between genes in a transcription mechanism from DNA microarray data. Collaboration is currently underway to interpret these results (personal communication with D. Wild and C. Rangel).