Low rank continuous-space graphical models

Carl Smith
Department of Chemistry
Columbia University
New York, NY 10027
cas2207@columbia.edu

Frank Wood
Department of Statistics
Columbia University
New York, NY 10027
fwood@stat.columbia.edu

Liam Paninski
Department of Statistics
Columbia University
New York, NY 10027
liam@stat.columbia.edu

Abstract

Constructing tractable dependent probability distributions over structured continuous random vectors is a central problem in statistics and machine learning. It has proven difficult to find general constructions for models in which efficient exact inference is possible, outside of the classical cases of models with restricted graph structure (chain, tree, etc.) and linear-Gaussian or discrete potentials. In this work we identify a tree graphical model class in which exact inference can be performed efficiently, owing to a certain "low-rank" structure in the potentials. We explore this new class of models by applying the resulting inference methods to neural spike rate estimation and motion-capture joint-angle smoothing tasks.

1 Introduction

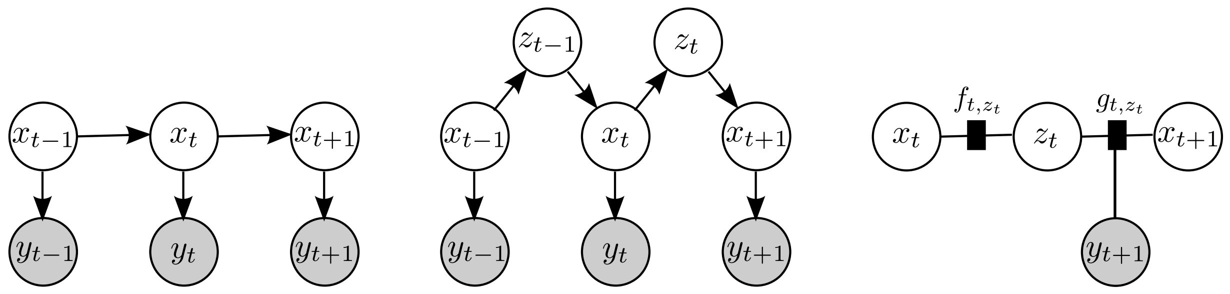
Graphical models make it easy to compose simple distributions into large, more expressive joint distributions. Unfortunately, in only a small subclass of graphical models is exact computation of marginals and conditionals relatively easy. In particular, while the problem of exact inference in discrete Markov random fields (MRFs) has seen a great deal of attention recently (Wainwright and Jordan, 2008), non-Gaussian MRFs defined on more general (non-discrete) state-spaces remain a more-or-less open challenge.

As a simple example, consider inference over a chain of dependent probabilities. Such a situation could, for instance, arise when modeling survey responses conducted over many years in which the same yes/no question is asked each year but where the data for some years are missing and of interest. One might want to estimate a population mean latent positive response probability for every year (including those years missing responses) that is expected to vary slowly from year to year. This requires specifying a smoothing prior on a sequence of variables that lie between between zero and one. There are many ways to specify such a smoothing prior, but even in this simple example it is hard to think of models that allow us to compute conditional expectations exactly and efficiently. (For example, the constraints on the latent variables and non-Gaussian likelihood rule out Kalman filtering in a transformed space.)

Similar to inferring latent sentiment in a survey response modeling application, one can find other latent variable "smoothing" tasks in fields as diverse as neuroscience and motion capture. In neuroscience, it is of interest to infer the latent probability of spiking – or firing rate – for a neuron given only observations of individual spikes over time. Note that this problem is very similar to the survey response problem above. We show results from "smoothing" neural firing probabilities to demonstrate the exact inference techniques proposed in this paper. We also show an example of smoothing motion capture joint angle time-series data, demonstrating that our exact method is applicable when even classical approximations break down.

The aim of this work is to expand the class of models for which exact inference is computationally feasible, in particular with models of continuous ans structured random variables with non-Gaussian densities. We start by reviewing an auxiliary variable method for introducing Markov chain dependencies between random variables of arbitrary type. We then develop an efficient method for exact inference in a subset of such models, and identify a new class of "low-rank" models in which exact inference is efficient. We perform inference on examples of such models.

Appearing in Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX. Copyright 2012 by the authors.



(a) Graphical model with "smoothing" dependency between latent variables.

(b) Graphical model with latent "smoothing" dependency induced by auxiliary variables.

(c) Factor graph corresponding to the graphical model in (b).

Figure 1

2 Related Work

To begin, we first review the work of Pitt et al. (2002) and Pitt and Walker (2005), who describe an auxiliary variable approach to introducing dependency between random variables of arbitrary types. Refer to Figure 1 and consider the sequence of random variables $X = \{x_t\}_{t=1}^T$. (Similarly we use the notation $Y = \{y_t\}_{t=1}^T$ and $Z = \{z_t\}_{t=1}^{T-1}$.) Assume that we would like to bias estimation of the x_t's such that for all values of t, $x_t \approx x_{t+1}$. For now, also assume that we would like the x's in this chain to be marginally identically distributed a priori, i.e. $x_t \sim G_0(x_t)$ for all t (this will be relaxed in later sections). One way to proceed is to require that G_0 is the invariant distribution of a Markov chain with transition kernel $p(x_t|x_{t-1})$, i.e. $G_0(x_t) = \int p(x_t|x_{t-1})G_0(x_{t-1})dx_{t-1}$. This constraint on $p(x_t|x_{t-1})$ is the same as that for any MCMC sampler of G_0; thus $p(x_t|x_{t-1})$ can be any valid sampler transition kernel, e.g. the Metropolis-Hastings transition kernel.

In Pitt and Walker (2005) a particular transition kernel based on the Gibbs sampler is considered. Their clever idea was to form a joint distribution $p(x, z)$ (dropping the subscript notation for the moment), defined as $p(x, z) = p(z|x)G_0(x)$. Clearly, if we Gibbs-sample from this distribution, i.e. sample $z_1 \sim p(z_1|x_1), x_2 \sim p(x_2|z_1), z_2 \sim p(z_2|x_2), \ldots$, then the marginal sequence $x_1, \ldots, x_T$ is marginally distributed as G_0, as desired. One advantage of this approach is that we have a great deal of freedom in our choice of $p(z|x)$. Pitt and Walker (2005) and others (Caron et al., 2007; Gasthaus et al., 2009) suggest choosing $p(z|x)$ to be conjugate to $G_0(x)$, since this implies that $p(x|z)$ is in the same family as G_0, making sampling more straightforward. In addition, as the amount of information in z about x is increased, neighboring values of x are more closely coupled together. We can also easily incorporate noisy observations y_t from this model (as shown in Figure 1): if the likelihood of y_t given x_t is also conjugate to G_0, then $p(x_t|z_{t-1}, y_t)$ remains in the same family as G_0, making conditional Gibbs sampling from $p(X|Y)$ straightforward.

3 Low-rank Markov chains

Constructing a Gibbs sampler to sample the x's and z's conditioned on observations (y's in Figure 1) is only asymptotically exact in the limit of infinite Gibbs sweeps. What has been overlooked until now (to our knowledge) is that the x's can often themselves be analytically marginalized out, leaving a Markov chain in the z's only, where computation often remains tractable when the z's are discrete random variables with a small state-space. Therefore, in the subset of this class of models in which the z's are discrete random variables, exact inference can be efficiently performed.

To see how this is possible, consider the form of the joint distribution of the graphical model in Figure 1b when the z's are discrete random variables. In this case we can write

$$p(X) = p(x_1) \prod_{t=1}^{T-1} \sum_{z_t} p(z_t|x_t)p(x_{t+1}|z_t)$$

where we disregard the observations y_t momentarily for the sake of clarity. To emphasize the primary role of the x's, $p(X)$ can be re-expressed in the following equivalent form

$$p(X) \propto \prod_{t=1}^{T-1} \sum_{z_t=1}^{R_t} f_{t,z_t}(x_t) g_{t,z_t}(x_{t+1}) \tag{1}$$

for appropriate functions f_{t,z_t} and g_{t,z_t}, where each sum is a potential coupling neighboring x variables, and where R_t is the size of the state-space of z_t, which we will refer to as the "rank" of the potential, for reasons that will become clear below. (The converse is also true; it is straightforward to show that, given nonnegative f_{t,z_t} and g_{t,z_t}, we can construct corresponding conditionals $p(z_t|x_t)$ and $p(x_{t+1}|z_t)$, although the resulting Markov chain in the x's may be non-stationary). In fact, the conditional distribution $p(X|Y)$ can be expressed in exactly the same form, by absorbing the observation densities $p(y_t|x_t)$ in the f or g terms. In Figure 1c we have chosen to include the y_t's in the g factor.

Now, if the x's were discrete random variables, then eq. (1) would represent a discrete Markov chain in which the transition matrices are of rank R_t. Recall that exact inference in such a low-rank Markov chain is relatively easy (Siddiqi et al., 2010), since the computational complexity of the forward-backward algorithm is dominated by the cost of multiplication by the transition matrix, and multiplication by low-rank matrices is relatively cheap.

The key idea is that, as long as the z's are discrete random variables with small state-space, exact inference on the Markov chain X defined in eq. (1) remains tractable. Even in the general (non-discrete X) case, exact inference requires just $O(R^2)$ time (assuming constant $R_t = R,\ t = 1, \dots, T$), as in a standard low-rank hidden Markov model; here the z's correspond to the latent variables. Consider the partition function of the joint distribution $p(X, Z)$.

$$\begin{aligned}
&\int dX_{1:T} \prod_{t=1}^{T-1} \sum_{z_t=1}^{R_t} f_{t,z_t}(x_t) g_{t,z_t}(x_{t+1}) \\
&\quad = \sum_{z_1=1}^{R_1} \left(\int dx_1 f_{1,z_1}(x_1) \int dx_2 g_{1,z_1}(x_2) \times \right. \\
&\quad \sum_{z_2=1}^{R_2} \left(f_{2,z_2}(x_2) \int dx_3 g_{2,z_2}(x_3) \cdots \right.
\end{aligned}$$

We arrive at the distribution of Z simply by removing the sums over the z's from the partition function. Rearranging the products and integrals above reveals the Markov structure of Z.

$$\begin{aligned}
p(Z) \propto &\int dx_1 f_{1,z_1}(x_1) \int dx_2 g_{1,z_1}(x_2) f_{2,z_2}(x_2) \times \\
&\int dx_3 g_{2,z_2}(x_3) f_{3,z_3}(x_3) \cdots \\
&\int dx_T g_{T-1,z_{T-1}}(x_T) \qquad (2)
\end{aligned}$$

We can use the forward-backward algorithm to compute exact marginals or samples from $p(Z)$; since, given Z, the x's are independent, we can therefore easily compute exact marginals or samples from $p(X)$ as well. To be explicit, expressions for the forward and backward variables are as follows:

$$\begin{aligned}
A_1^{(z_1)} &= \int dx_1 f_{1,z_1}(x_1) \\
A_t^{(z_t)} &= \sum_{z_{t-1}}^{R_{t-1}} A_{t-1}^{(z_{t-1})} \int dx_t g_{t-1,z_{t-1}}(x_t) f_{t,z_t}(x_t) \\
B_T^{(z_{T-1})} &= \int dx_T g_{T-1,z_{T-1}}(x_T) \\
B_t^{(z_{t-1})} &= \sum_{z_t}^{R_t} B_{t+1}^{(z_t)} \int dx_t g_{t-1,z_{t-1}}(x_t) f_{t,z_t}(x_t) \qquad (3)
\end{aligned}$$

These are message passing equations (Bishop, 2006) and the forward and backward variables can be computed by induction on t. Given these quantities, the marginal distributions of the X become mixture of modes indexed by the z_t:

$$p(x_t) \propto \sum_{z_{t-1}}^{R_{t-1}} A_{t-1}^{(z_{t-1})} \sum_{z_t}^{R_t} B_{t+1}^{(z_t)} g_{t-1,z_{t-1}}(x_t) f_{t,z_t}(x_t)$$

So, if the inner products $\int g_{t-1,i}(x) f_{t,j}(x) dx$ can be evaluated then we can perform exact inference in X (or more generally in X given observations Y) in $O\left(\sum_{t=1}^T R_t^2\right)$ time, by the forward-backward algorithm sketched above. (Note that we need only compute these inner products once; these can therefore be pretabulated if necessary before inference begins.) It is straightforward to show that the linear scaling of this inference with T holds for general acyclic Markov random fields (i.e., trees) with potentials of the low-rank form described in eq. (1). Moreover, for certain graphs with cycles, the full $p(X)$ or $p(X|Y)$ can be treated efficiently as a weighted sum of trees via the method of conditioning (Pearl, 1988).

When applying such a model to data, it will usually not be the case that we know the rank of the potential functions f and g. In this case R has to be estimated from data. This is a standard model selection problem; a Bayesian approach would exploit the marginal likelihood $p(Y|R) = \int p(X|R)p(Y|X)dX$ of the observed data Y given the rank R. This marginal likelihood can be computed directly from our forward recursion (as usual in the context of hidden Markov models (Rabiner, 1989)); see Fig. 2 for an illustration.

Finally, Z is guaranteed to be a proper Markov chain only if all the inner products over f and g are positive. On the other hand, mathematically there is nothing against performing the recursive inference with the above forward-backward variables when the inner products can be negative, though numerical issues due to cancellation of numbers below machine precision may be a problem in this case. We will stick to nonnegative potentials in this work.

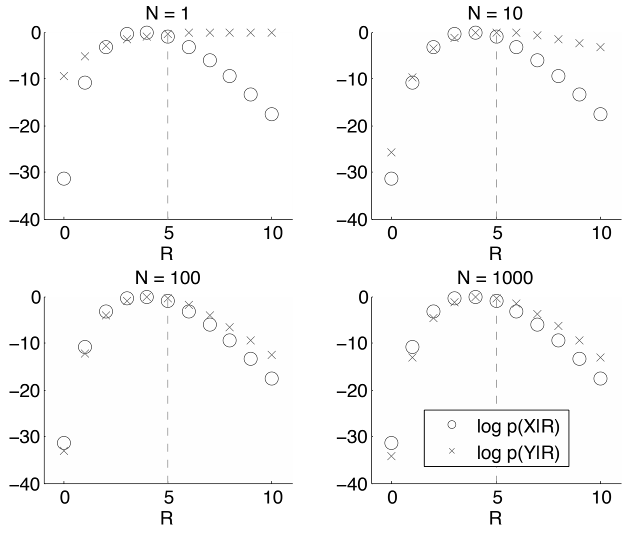


Figure 2: The marginal likelihood can be used to estimate the rank of the underlying process X generating data Y. Here a sample X_0 was generated from the beta-binomial time series model $p(X|R)$ (Section 4.1) with rank $R = 5$; i.e., $R_t = 5$ for all times t. We plot the log-likelihood (circles) $\log p(X_0|R)$ as a function of R. Then we generate data Y from $p(Y|X_0)$ and use the data to estimate the rank by maximizing the log-likelihood $\log p(Y|R)$ (crosses) as a function of R. As the binomial parameter $N_t \equiv N$ increases, the data Y become more informative about X_0, and $p(Y|R)$ approaches $p(X_0|R)$.

4 Examples

4.1 Beta-binomial and Dirichlet-multinomial time series

We now return to the probability-smoothing example we mentioned in the introduction. We consider a time series of binomial distributed data $y_t \sim \text{Binomial}(N_t, x_t)$. If we choose any prior $p(X)$ such that the posterior $p(X|\{N_t\}, Y)$ has the form of eq. (1), then exact inference is tractable. For example, we could choose x_t and z_t to have the following simple conjugate Beta-binomial form:

$$\begin{aligned} x_1 &\sim \text{Beta}(\alpha, \beta) \\ z_t|x_t &\sim \text{Binomial}(z_t; R_t, x_t) \\ x_{t+1}|z_t &\sim \text{Beta}(\alpha + z_t, \beta + R_t - z_t) \end{aligned}$$

Thus x_t is marginally Beta(α, β), and the dependence between x_t and x_{t+1} — i.e., the smoothness of the x's as a function of time — is set by R_t: large values of R_t lead to strongly-coupled x_t and x_{t+1}. Eq. (1) in this case becomes

$$p(X) = \prod_{t=1}^{T-1} \sum_{z_t=0}^{R_t} a_{z_t} x_t^{z_t} (1 - x_t)^{R_t - z_t} \times x_{t+1}^{z_t} (1 - x_{t+1})^{R_t - z_t} \quad (4)$$

which we will call the beta-binomial smoother, for the appropriate coefficients a_{z_t}.

We could consider more general priors of the form

$$p(X) \propto \prod_t \sum_{i=0}^{R_t} a_{ti} x_t^{\alpha_i} (1 - x_t)^{\beta_i} x_{t+1}^{\gamma_i} (1 - x_{t+1})^{\delta_i}$$

where α_i, β_i, γ_i, and δ_i are greater than or equal to -1 so that the inner product integrals don't diverge, and $a_{ti} > 0$ for the reasons described above. Given the form of the binomial likelihood, that the posterior $p(X|\{N_t\}, Y)$ will have the same form, but with the constants α_i, β_i, γ_i, and δ_i modified accordingly. Distributions of this form could be considered as tractable conjugate priors for binomial time series data. Note that the necessary inner products can be computed easily in terms of standard beta functions, and inference proceeds in $O(R^2)$ time, assuming constant $R_t = R$.

Multivariate generalizations are conceptually straightforward: we replace beta distributions with Dirichlets and binomials by multinomials, since by analogy to the beta-binomial model, the Dirichlet is conjugate to the multinomial distribution:

$$\begin{aligned} \vec{x}_1 &\sim \text{Dirichlet}(\vec{\alpha}) \\ \vec{z}_t|\vec{x}_t &\sim \text{Multinomial}(R_t, \vec{x}_t) \\ \vec{x}_t|\vec{z}_t &\sim \text{Dirichlet}(\vec{\alpha} + \vec{z}_t) \\ \vec{y}_t|\vec{x}_t, n_t &\sim \text{Multinomial}(\vec{y}_t; \vec{x}_t, n_t) \end{aligned}$$

Just as in the beta-binomial case, this defines a sequence of marginally-Dirichlet distributed probabilities x_t, with R_t controlling the smoothness of the state path X. Inference in this case scales quadratically with the total number of possible histograms $\vec{z}_t$ that might be observed.

4.2 Smoothing conjugate priors for multinomial data

In many cases one would like a conjugate prior for multinomial data that leads to smooth estimates of the underlying probabilities. In the preceding example, we constructed a conjugate prior for count data that has smooth and nonnegative sample paths. If we further constrain these sample paths to sum to one, then we could interpret X as a discrete probability

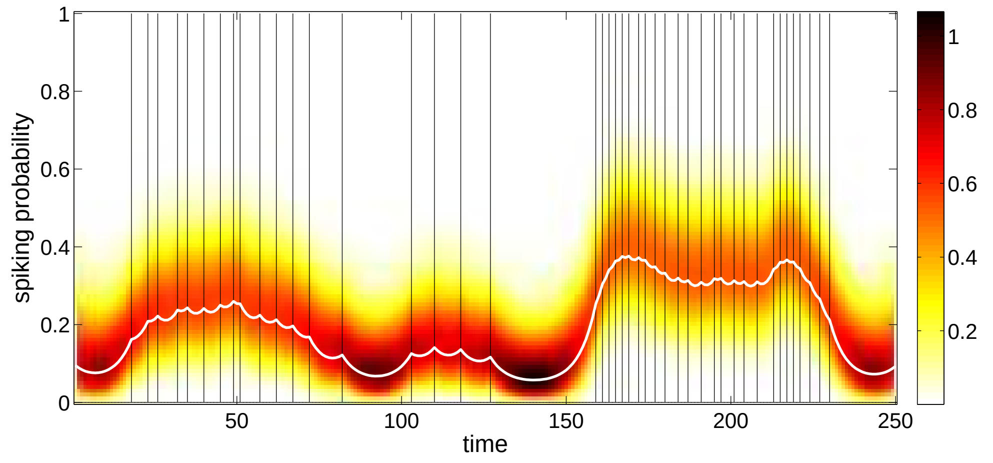


Figure 3: The inferred spiking probability density from spike train data assuming a binomial spiking model and the beta-binomial smoother. The black bars are the observed spikes. The solid white line is the inferred mean of the spiking probability. Each time unit is 2 ms.

distribution; it is easy to see that the resulting smoothing prior $p(X)$ is conjugate to multinomial data, due to the completely factorized form of the multinomial likelihood. However, it is not immediately clear how to exploit the model's low-rank structure to perform inference in a tractable way, since the constraint that the components of X sum to one breaks the tree structure of the graphical model.

One approach is to transform to a larger state-space, $x_t \to q_t = (x_t \;\; s_t)$, where s_t denotes the cumulative sum $s_t = x_1 + x_2 + \cdots + x_t$. This leads to a Markov prior on the augmented state variable Q of the form

$$\begin{aligned} p(Q) \propto \delta(s_1 = x_1) x_1^{\nu_1 - 1} \sum_{k_1}^{R} a_{k_1} x_1^{k_{1,1}} x_2^{k_{1,2}} \times \\ \delta(s_1 = s_2 - x_2) x_2^{\nu_2 - 1} \sum_{k_2}^{R} a_{k_2} x_2^{k_{2,1}} x_3^{k_{2,2}} \times \\ \cdots \delta(s_{T-2} = s_{T-1} - x_{T-1}) x_{T-1}^{\nu_{T-1} - 1} \times \\ \sum_{k_{T-1}}^{R} a_{k_{T-1}} x_{T-1}^{k_{T-1,1}} x_T^{k_{T-1,2}} \times \qquad (5) \\ \delta(s_{T-1} = s_T - x_T) x_T^{\nu_T} \delta(s_T = 1) \end{aligned}$$

As outlined in greater detail in the appendix, we can perform forward-backward inference on this density by recursively integrating the above density, over all the x_i and the s_i, to compute the normalization constant, and then rearranging the summations into the form of a sum-product algorithm. The resulting inference algorithm requires $O(R^2T^2)$ storage and $O(R^6T^2)$ processing time.

4.3 Phase data

So far our random variables have lived in convex subsets of vector spaces; standard approximation methods (e.g., Laplace approximation (Kass and Raftery, 1995) or expectation propagation (Minka, 2001)) can often be invoked to perform approximate inference in these settings. However, our method may be applied on more general state-spaces, where these classical approximations break down. As a concrete example, consider a time-series of phase variables (angles). The von Mises distribution

$$p(x_t|\mu_t, \kappa_t) \propto e^{\kappa_t \cos(x_t - \mu_t)}$$

(with mean and concentration parameters μ_t and κ_t) is popular for modeling one-dimensional angular data, largely because the necessary normalization factors can be computed easily, and furthermore this model has the convenient feature that, like the normal density, it is conjugate to itself (Gelman et al., 2003). As in our previous examples, this univariate distribution can be augmented to tractably model smoothed time-series data. For instance, we could take

$$p(X) \propto \prod_t \sum_{i=0}^{R} e^{\frac{R}{2} \cos\left(x_t - \frac{2\pi i}{R+1}\right)} e^{\frac{R}{2} \cos\left(x_{t+1} - \frac{2\pi i}{R+1}\right)} \qquad (6)$$

This acts as a smoothing prior, since at each time t, for each corresponding pairwise potential, each of the terms in the sum over i is a unimodal function peaked at $x_t = x_{t+1} = \frac{2\pi i}{R+1}$. That is, each term contributes a bump along the diagonal, and therefore the sum over i corresponds to a nearly-diagonal transition matrix, i.e. to a smoothing prior. Larger values of R lead to smoother sample paths in X. Inference proceeds as in the previous examples; if the observations y_t also have von Mises densities given x_t (as in the example application discussed in the next section), then the necessary inner products can be computed easily in terms of Bessel functions.

As in the Dirichlet-multinomial case, extensions to multivariate phase data are conceptually straightforward (the von Mises-Fisher density generalizes the univariate von Mises density (Mardia and Jupp, 2000); see Cadieu and Koepsell (2010) for another generalization). We will describe another generalization, to oscillatory or narrowband time series data, below.

5 Experiments

We began by analyzing some simple neural spike train data (http://neurotheory.columbia.edu/~larry/book/exercises.html) using the beta-binomial smoother. A segment of a spike train in which each time unit represents 2 ms was obtained. The spikes (the binary observations $\{y_t\}$) were modeled as draws from a binomial distribution with time-varying probability x_t. The smoother of eq. (4) was used with $\alpha = \beta = 1$, setting the a priori marginals to be uniform distributions. We used $R = 100$, which leads to a prior autocorrelation time of approximately 60 ms. The forward-backward algorithm was run to infer the distribution over x_t as a function of time as shown in Figure 3. The marginal mean varies smoothly over time, rising during times of higher spike rates.

We also performed some basic comparisons to Gibbs sampling. The Gibbs sampler is the standard approach to computation in this type of model, but as emphasized above it only leads to approximate solutions, whereas the marginalized forward-backward approach we have introduced here provides exact results. The basic result, shown in Figure 4 is unsurprising: many Gibbs sweeps are required to achieve a certain error level, particularly in cases where the sample paths from the conditional distribution $p(X|Y, R)$ are strongly coupled.

Next we turned to a dataset involving phase variables. We analyzed joint articulation motion capture data from the CMU Graphics Lab Motion Capture Database (http://mocap.cs.cmu.edu/search.php?subjectnumber=13&trinum=9). A time-series of angles of extension of the right radius of a man drinking from a bottle of soda was analyzed. This motion was modeled with the von Mises smoother of eq. (6) with $R = 20$ and $\kappa = 2$. The observations y_t were modeled as von Mises draws with mean x_t. The forward-backward algorithm smoothed the data effectively and allowed for appropriate inference in the presence of missing data, as illustrated in Fig. 5.

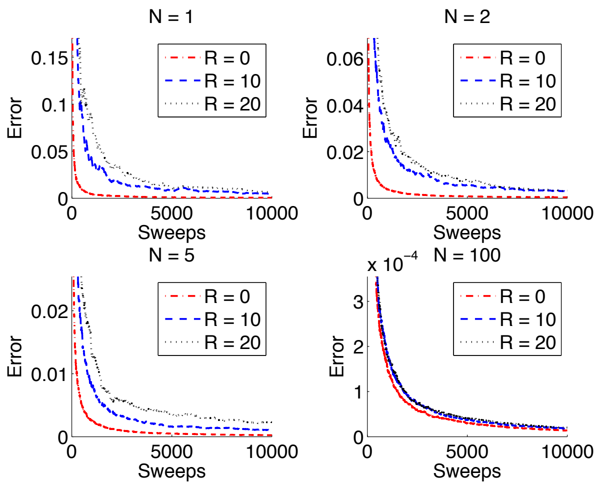


Figure 4: For different amounts N of data, the marginal means of a Markov chain X of length $T = 100$ were computed both exactly and approximately by Gibbs sampling, using the beta-binomial smoother. Here we plot the root mean square error per time step of the Gibbs solution with respect to the exact solution as a function of the number of Gibbs sweeps. For each value N of the binomial count parameter we plot this curve for three values of the rank R. Each curve is the median of 25 traces, each the average of 10 independent runs of the Gibbs sampler. Each of the 25 traces corresponds to different randomly generated input data from $p(Y|R)$. The sampler was initialized with each x_t drawn independently from the marginal prior distribution, $p(x_t) = Uniform([0, 1])$. The Gibbs estimates converge most quickly when $p(X|Y, R)$ is most uncoupled, that is, when R is small and/or N is large; when R is large or N is small the Gibbs error requires many sweeps to shrink towards zero.

Conceptually, we are applying a rather simple state-space model to this data, with the true underlying angle (the hidden state variable) modeled as $x_{t+1} = x_t + \epsilon_t$, and the observation modeled as $y_t = x_t + \eta_t$ for appropriate noise terms ϵ_t and η_t. This state-space viewpoint suggests some natural further generalizations. For example, if we let $x_{t+1} = x_t + 2\pi\omega + \epsilon_t$, then x_t could model a narrowband signal with dominant frequency ω. Our inference methods can be applied in a straightforward manner to this oscillatory model, and may therefore be useful in a number of potential applications, e.g. the analysis of noisy electroencephalography data, or in the acoustic applications described in Turner and Sahani (2011).

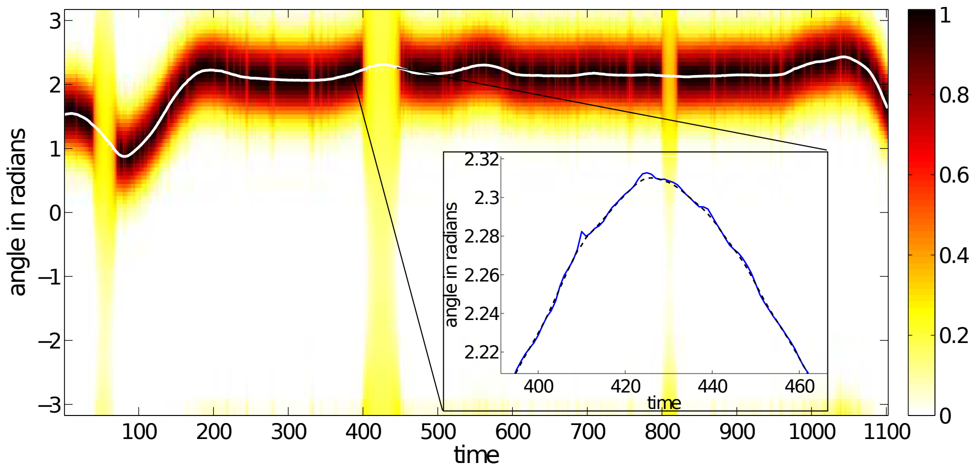


Figure 5: The inferred probability density of angle in motion capture data. The solid white line is the observed signal. The dotted white line, mostly obscured by the solid white line, is the inferred mean. The colorbar indicates the inferred posterior. Bands appear in intervals where the observations are suppressed. They are tapered because, deeper within the band, distant observations are less informative of the density, which is therefore nearly uniform. Inset: Inference with no data held out. The solid line is the observed signal, the dashed line the inferred mean. Much of the noise in the signal has been smoothed. Each time unit is 2 ms.

6 Discussion

We have introduced a class of "low-rank" models for continuous-valued data in which exact inference is possible by efficient forward-backward methods. These exactly-solvable models are perhaps of most interest in cases where standard approximation methods (e.g., expectation propagation or Laplace approximation) are unreliable, such as the application to circular data time series discussed in section 4.3. Even in less "exotic" cases, such as the beta-binomial model discussed in section 4.1, classical methods based on Gibbs sampling can mix slowly (c.f. Fig. 4), making the exact sampler introduced here more attractive. (More generally, of course, there is significant value in exact, not approximate, inference methods: in mission-critical applications, for example, it is essential to have methods that are guaranteed to return the correct answer 100% of the time.) Thus we hope that these low-rank models might prove useful in a wide range of applications.

Directions for future theoretical research include connections with recent work on inference in reproducing kernel Hilbert spaces (RKHSs) by Song et al. (2010a) and Song et al. (2010b). The latter describe an approximate RKHS inference method in tree-structured graphical models, e.g. over non-Gaussian continuous variables. They perform belief propagation by operations on messages represented in a RKHS. Their approximate inference algorithm takes as input samples from the variables and estimates the necessary operators. It could be fruitful to explore connections between such a program and the methods we present here.

Our models also bear resemblance to the Reduced-Rank Hidden Markov Models (RR-HMMs) proposed by Siddiqi et al. (2010). These are n-state hidden Markov models with rank $k < n$ transition matrices. Probabilities in RR-HMMs can be represented in terms of $k \times k$ matrices, and inference can be done in $O(k^2)$ time. Our method exploits a similar property of the models it treats, reformulating transitions between continuous state-spaces (not only between high-dimensional discrete state-spaces) in terms of low-dimensional discrete spaces.

Further, there are a number of models that include inference in a large number of chains of dependent, constrained random variables for which our exact inference approach might not only improve inference but may result in significant computational savings. One example is the generalized Polya-urn dependent Dirichlet process (GPU-DDP) mixture (Caron et al., 2007). The GPU-DDP models time series observations as being draw from a time-dependent Dirichlet mixture. The latent parameters of the mixture components are allowed to change over time, but must be constrained in the same way that the auxiliary variable random walk of Pitt et al. (2002) constrains the latent sample paths in this paper. Inference in GPU-DDP mixtures is hard, suffering from slow mixing and high computational complexity, particularly in the low sample count, high-rank domain in which our exact inference approach excels. Applying our inference procedure to GPU-DDP inference could result in substantial improvements.

7 Appendix

Here we derive forward variables (FV) of the multinomial data smoother of section 4.2. A detailed derivation is available online. Let $\{a_k(t)\} = \{a_k\}$ be coefficients, indexed by multi-index k, assumed here to be constant over time. Define $s_0 = 0$ and $\delta_i \equiv \delta(s_i = s_{i+1} - x_{i+1})$. Deriving FV, the exponents $\{\nu_i - 1\}$ in (5) may be considered absorbed by the $k_{i,j}$, to simplify notation. We compute the normalization constant:

$$\begin{aligned}\mathcal{Z} &= \int_0^1 ds_1 \int_0^{s_1} dx_1 \cdots \int_0^1 ds_T \int_0^{s_T} dx_T\, p(Q) \\ &= \int_0^1 ds_1 \int_0^{s_1} dx_1 \sum_{k_1}^{R} a_{k_1} x_1^{k_{1,1}} \delta_0 \times \\ &\int_0^1 ds_2 \int_0^{s_2} dx_2 \sum_{k_2}^{R} a_{k_2} x_2^{k_{1,2}+k_{2,1}} \delta_1 \times \cdots\end{aligned}$$

We integrate left to right, first dx_1, then ds_1, dx_2, ds_2, and so on through ds_T. Integration over x_1 removes δ_0, leaving $s_1^{k_{1,1}}$ in place of $x_1^{k_{1,1}}$. Integration over s_1 removes δ_1, leaving $(s_2 - x_2)^{k_{1,1}}$ instead. Integration over x_2 proceeds by a change of variables:

$$\int_0^s dx (s-x)^{\alpha-1} x^{\beta-1} \overset{u=x/s}{=} s^{\alpha+\beta-1} \mathrm{B}(\alpha, \beta)$$

where $\mathrm{B}(\alpha,\beta)$ is the beta function. Integration over s_2 removes δ_2, leaving $(s_3 - x_3)^{k_{1,1}+k_{1,2}+k_{2,1}}$. Integration over x_3 uses the same variable change. Continuing, $\mathcal{Z}$ becomes the following nested sum:

$$\mathcal{Z} = \sum_{k_1}^{R} a_{k_1} \left(\sum_{k_2}^{R} a_{k_2} b(k_{1:2}) \left(\sum_{k_3}^{R} a_{k_3} b(k_{1:3}) \cdots \left(\sum_{k_{T-1}}^{R} a_{k_{T-1}} b(k_{1:T-1}) b(k_{1:T}) \right) \cdots \right)\right)$$

where we define $k_{0,2} \equiv k_{T,1} \equiv 0$, $K_i \equiv \sum_{j=1}^{i} k_{j-1,2} + k_{j,1}$, and $b(k_{1:n}) \equiv \mathrm{B}(k_{n-1,2} + k_{n,1}, K_{n-1})$.

This is not in sum-product form because, e.g., $b(k_{1:T})$ depends on all the indices of summation. We may put this into sum-product form as follows. A sum over k_i is a sum over $k_{i,1} \in \{0, \cdots, R\}$, then a sum over $k_{i,2} \in \{0, \cdots, R\}$. Notice that every set of values $\{k_i\}_{i=1}^{T} = \{[k_{i,1}, k_{i,2}]\}_{i=1}^{T}$ corresponds uniquely to a set of values $\{[k_{i,1}, K_i]\}_{i=1}^{T}$, and vice versa. Then we may sum over the latter in the order K_T, $k_{T-1,1}$, K_{T-1}, $k_{T-2,1}$, and so on, instead of summing over k_i, with the same result. The index values must be constrained; $k_{1,1} = 1$ is not compatible with $K_2 = 0$, because $K_2 = k_{1,1} + k_{1,2} + k_{2,1} \geq k_{1,1}$. This requirement manifests in the upper and lower bounds of the sums over these indices. It can be shown that $k_{i,1} \in \{\max\{0, K_{i+1} - k_{i+1,1} - (2i-1)R\}, \cdots, \min\{R, K_{i+1} - k_{i+1,1}\}\}$ and $K_i \in \{\max\{k_{i,1}, K_{i+1} - k_{i+1,1} - R\}, \cdots, K_{i+1} - k_{i+1,1}\}$:

$$\mathcal{Z} = \sum_{K_T=0}^{R(2T-2)} \left(\sum_{k_{n-1,1}} \sum_{K_{n-1}} \cdots \left(\sum_{k_{2,1}} \sum_{K_2} a_{[k_{2,1}, K_3 - K_2 - k_{3,1}]} \mathrm{B}(K_3 - K_2, K_2) \left(\sum_{k_{1,1}} a_{[k_{1,1}, K_2 - K_1 - k_{2,1}]} \mathrm{B}(K_2 - K_1, K_1) \right)\right) \cdots \right)$$

The FV, then, are as follows. The first and second superscripts index values of $k_{t,1}$ and K_t, respectively:

$$A_2^{(i,j)} = \sum_{k=\max\{0, j-i-R\}}^{\min\{R, j-i\}} a_{[k, j-k-i]} \mathrm{B}(j-k, k)$$

$$i \in \{0, \cdots, R\}, j \in \{i, \cdots, i+2R\}$$

$$A_t^{(i,j)} = \sum_{k=\max\{0, j-i-(2t-3)R\}}^{\min\{R, j-i\}} \sum_{l=\max\{k, j-i-R\}}^{j-i} a_{[k, j-l-i]} \mathrm{B}(j-l, l) A_{t-1}^{(k,l)}$$

$$i \in \{0, \cdots, R\}, j \in \{i, \cdots, i+2(t-1)R\}$$

$$A_T^{(j)} = \sum_{k=\max\{0, j-(2t-4)R\}}^{\min\{R, j\}} \sum_{l=\max\{k, j-R\}}^{j} a_{[k, j-l]} \mathrm{B}(j-l, l) A_{T-1}^{(k,l)}$$

$$j \in \{0, \cdots, 2(T-1)R\}$$

and $\mathcal{Z} = \sum_{i=0}^{2(T-1)R} A_T^{(i)}$. Similarly we can derive backward variables $C_t^{(i,j)}$, where the first superscript indexes $k_{t-1,2}$ and the second indexes $L_t \equiv \sum_{i=t}^{T} k_{i-1,2} + k_{i,1}$. Marginals can be computed readily:

$$p(x_t) = \frac{1}{\mathcal{Z}} \sum_{i=0}^{R} \sum_{j=i}^{i+2(t-2)R} \sum_{k=0}^{R} \sum_{l=k}^{k+2(T-1-t)R} A_{t-1}^{(i,j)} C_{t+1}^{(k,l)} \mathrm{B}(j,l) \sum_{k_{t-1,2}=0}^{R} \sum_{k_{t,1}=0}^{R} a_{[i, k_{t-1,2}]} a_{[k_{t,1}, k]} x_t^{k_{t-1,2}+k_{t,1}} (1-x_t)^{j+l} \tag{7}$$

This method requires $O(R^2T^2)$ storage for the FV, and $O(R^6T^2)$ time to compute marginals. For time, one factor of T comes from the number of marginals to compute, the other from the number of $A_t^{(i,j)}$ for each t, which increases linearly with T, manifesting in the limits of the sums over j and l in eq. (7).

References

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Cadieu, C. F. and Koepsell, K. (2010). Phase coupling estimation from multivariate phase statistics. *Neural Computation*, 22(12):3107–3126.

Caron, F., Davy, M., and Doucet, A. (2007). Generalized Polya urn for time-varying Dirichlet process mixtures. In *Proc. 23rd Conf. on Uncertainty in Artificial Intelligence (UAI)*.

Gasthaus, J., Wood, F., Görür, D., and Teh, Y. W. (2009). Dependent Dirichlet process spike sorting. In *Advances in Neural Information Processing Systems*, pages 497–504.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian Data Analysis*. Chapman and Hall/CRC, 2 edition.

Kass, R. and Raftery, A. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Mardia, K. and Jupp, P. (2000). *Directional statistics*. Wiley series in probability and statistics. Wiley.

Minka, T. (2001). *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, MIT.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc.

Pitt, M. K., Chatfield, C., and Walker, S. G. (2002). Constructing first order stationary autoregressive models via latent processes. *Scandinavian Journal of Statistics*, 29(4):657–663.

Pitt, M. K. and Walker, S. G. (2005). Constructing stationary time series models using auxiliary variables with applications. *Journal of the American Statistical Association*, 100(470):554–564.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Siddiqi, S. M., Boots, B., and Gordon, G. J. (2010). Reduced-rank hidden Markov models. In *Proc. 13th Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*.

Song, L., Boots, B., Siddiqi, S. M., Gordon, G. J., and Smola, A. J. (2010a). Hilbert space embeddings of hidden Markov models. In *Proc. 27th Intl. Conf. on Machine Learning*.

Song, L., Gretton, A., and Guestrin, C. (2010b). Nonparametric tree graphical models via kernel embeddings. In *Proc. 13th Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*.

Turner, R. E. and Sahani, M. (2011). Probabilistic amplitude and frequency demodulation. In *Advances in Neural Information Processing Systems*, pages 981–989.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305.

Extended appendix

In section 4.2 we describe a polynomial time smoother for multinomial data confined to the unit simplex. Here we derive the forward variables of this smoother, which we need to perform efficient inference. The necessary backward variables may be derived following the same integration steps but starting from time T and proceeding backward.

Define $s_0 \equiv 0$, and let k_i be the multi-index $[k_{i1}, k_{i2}]$, over which the sum $\sum_{k_i} x_i^{k_{i1}} x_{i+1}^{k_{i2}}$ couples x_i to x_{i+1} (in addition to the implicit coupling by the simplex constraint). Let $\{a_k(t)\} = \{a_k\}$ be a set of coefficients, indexed by multi-index k, that we assume to be constant over time, merely for notational convenience. Pushing all sums as far to the right as possible, and defining $\delta_i \equiv \delta(s_i = s_{i+1} - x_{i+1})$, the joint density in the expanded state-space is

$$
\begin{aligned}
p(Q) = &\sum_{k_1}^{R} a_{k_1} x_1^{\nu_1 + k_{1,1} - 1} \delta_0 \times \\
&\sum_{k_2}^{R} a_{k_2} x_2^{\nu_2 + k_{1,2} + k_{2,1} - 1} \delta_1 \times \cdots \\
&\sum_{k_{T-1}}^{R} a_{k_{T-1}} x_{T-1}^{\nu_{T-1} + k_{T-2,2} + k_{T-1,1} - 1} \delta_{T-2} \times \\
&x_T^{\nu_T + k_{T-1,2} - 1} \delta_{T-1} \delta(s_T = 1)
\end{aligned}
$$

For the purpose of computing the forward variables, the multinomial exponents $\{\nu_i - 1\}$ may be omitted and considered absorbed by the $k_{i,j}$, so as to further simplify the notation.

We compute the normalization constant of the joint distribution:

$$
\begin{aligned}
\mathcal{Z} &= \int_0^1 ds_1 \int_0^{s_1} dx_1 \cdots \int_0^1 ds_T \int_0^{s_T} dx_T \, p(Q) \\
&= \int_0^1 ds_1 \int_0^{s_1} dx_1 \sum_{k_1}^{R} a_{k_1} x_1^{k_{1,1}} \delta_0 \times \\
&\int_0^1 ds_2 \int_0^{s_2} dx_2 \sum_{k_2}^{R} a_{k_2} x_2^{k_{1,2} + k_{2,1}} \delta_1 \times \cdots
\end{aligned}
$$

To compute $\mathcal{Z}$, we integrate from left to right, first integrating dx_1, then ds_1, then dx_2, then ds_2, and so on until ds_T. Even after only the first five integrations, a pattern begins to emerge:

$$
\begin{aligned}
\mathcal{Z} = &\sum_{k_1} a_{k_1} \int_0^1 ds_1 (s_1 - s_0)^{k_{1,1}} \times \\
&\int_0^1 ds_2 \int_0^{s_2} dx_2 \sum_{k_2} a_{k_2} x_2^{k_{1,2} + k_{2,1}} \delta_1 \times \cdots
\end{aligned}
$$

$$
\begin{aligned}
= &\sum_{k_1} a_{k_1} \int_0^1 ds_2 \int_0^{s_2} dx_2 (s_2 - x_2)^{k_{1,1}} \times \\
&\sum_{k_2} a_{k_2} x_2^{k_{1,2} + k_{2,1}} \times \cdots
\end{aligned}
$$

$$
\begin{aligned}
= &\sum_{k_1} a_{k_1} \sum_{k_2} a_{k_2} \mathrm{B}(k_{1,2} + k_{2,1}, k_{1,1}) \times \\
&\int_0^1 ds_2 s_2^{k_{1,1} + k_{1,2} + k_{2,1}} \times \\
&\int_0^1 ds_3 \int_0^{s_3} dx_3 \sum_{k_3} a_{k_3} x_3^{k_{2,2} + k_{3,1}} \delta_2 \times \cdots
\end{aligned}
$$

$$
\begin{aligned}
= &\sum_{k_1} a_{k_1} \sum_{k_2} a_{k_2} \mathrm{B}(k_{1,2} + k_{2,1}, k_{1,1}) \times \\
&\int_0^1 ds_3 \int_0^{s_3} dx_3 (s_3 - x_3)^{k_{1,1} + k_{1,2} + k_{2,1}} \times \\
&\sum_{k_3} a_{k_3} x_3^{k_{2,2} + k_{3,1}} \times \cdots
\end{aligned}
$$

$$
\begin{aligned}
= &\sum_{k_1} a_{k_1} \sum_{k_2} a_{k_2} \mathrm{B}(k_{1,2} + k_{2,1}, k_{1,1}) \times \\
&\sum_{k_3} a_{k_3} \mathrm{B}(k_{2,2} + k_{3,1}, k_{1,1} + k_{1,2} + k_{2,2}) \times \\
&\int ds_3 s_3^{k_{1,1} + k_{1,2} + k_{2,1} + k_{2,2} + k_{3,1}} \times \\
&\int_0^1 ds_4 \int_0^{s_4} dx_4 \sum_{k_4} a_{k_4} x_4^{k_{3,2} + k_{4,1}} \delta_3 \times \cdots
\end{aligned}
$$

The first equality results from integration with respect to x_1, which removes the delta function $\delta_0 = \delta(s_0 = s_1 - x_1) = \delta(x_1 = s_1)$, leaving $(s_1 - s_0)^{k_{1,1}}$ in place of $x_1^{k_{1,1}}$. The second equality results from integration with respect to s_1, which removes the delta function $\delta_1 = \delta(s_1 = s_2 - x_2)$, which replaces $(s_1 - s_0)^{k_{1,1}} = s_1^{k_{1,1}}$ with $(s_2 - x_2)^{k_{1,1}}$. Next we integrate with respect to x_2, which proceeds by the following change of variables:

$$
\begin{aligned}
&\int_0^s dx (s - x)^{\alpha - 1} x^{\beta - 1} \overset{u = x/s}{=} \\
&\int_0^1 du (s(1 - u))^{\alpha - 1} (su)^{\beta - 1} s = \\
&s^{\alpha + \beta - 1} \int_0^1 (1 - u)^{\alpha - 1} u^{\beta - 1} = s^{\alpha + \beta - 1} \mathrm{B}(\alpha, \beta)
\end{aligned}
$$

where $\mathrm{B}(\alpha, \beta)$ is the beta function. The next integration with respect to s_2 removes the delta function $\delta_2 = \delta(s_2 = s_3 - x_3)$, resulting in the power of $(s_3 - x_3)$. The last equality is a result of the same change of variables when integrating with respect to x_3, yielding another beta function, and leaving the last line of the expression identical to the last line of the expression two steps before, except that the "time" indices have all increased by one.

To condense this expression, define $k_{0,2} \equiv k_{T,1} \equiv 0$, and $K_i \equiv \sum_{j=1}^{i} k_{j-1,2} + k_{j,1}$. Lastly, define $b(k_{1:n}) \equiv \mathrm{B}(k_{n-1,2} + k_{n,1}, K_{n-1})$. Continuing with the integration from left to right, we find an expression for the normalization constant as the following nested sum:

$$\mathcal{Z} = \sum_{k_1}^{R} a_{k_1} \left(\sum_{k_2}^{R} a_{k_2} b(k_{1:2}) \left(\sum_{k_3}^{R} a_{k_3} b(k_{1:3}) \times \cdots \right.\right.$$
$$\left(\sum_{k_{T-2}}^{R} a_{k_{T-2}} b(k_{1:T-2}) \left(\sum_{k_{T-1}}^{R} a_{k_{T-1}} b(k_{1:T-1}) \times \right.\right.$$
$$\left.\left.\left.\left. b(k_{1:T}) \right)\right) \cdots \right)\right)$$

This is not in sum-product form because, e.g., $b(k_{1:T})$ depends on all the indices of summation. We may put this into sum-product form as follows. However, we can rearrange this summation into the form of a tractable sum-product algorithm as follows. Each sum over k_i is first a sum over $k_{i,1} \in \{0, \cdots, R\}$, and then a sum over $k_{i,2} \in \{0, \cdots, R\}$. The sums are in the order $k_{1,1}$, $k_{1,2}$, $k_{2,1}$, $k_{2,2}$, and so on. Now, notice that every set of values $\{k_i\}_{i=1}^T = \{[k_{i,1}, k_{i,2}]\}_{i=1}^T$ corresponds uniquely to a set of values $\{[k_{i,1}, K_i]\}_{i=1}^T$, with K_i as defined above, and vice versa. Then we may sum over values of the latter quantity in the order K_T, $k_{T-1,1}$, K_{T-1}, $k_{T-2,1}$, and so on, instead of summing over values of k_i, and obtain the same result. That is, we treat the K_i as sum indices instead of explicitly summing over the $k_{i,2}$.

To sum over all possible values of this second set of indices, the values of the indices must be constrained to be compatible. For instance, $k_{1,1} = 1$ is not compatible with $K_2 = 0$, because $K_2 = k_{1,1} + k_{1,2} + k_{2,1} \geq k_{1,1}$. This compatibility requirement manifests in the upper and lower bounds of the sums over these indices in eq. (8). Consider the sum over $k_{1,1}$, given some values for $k_{2,1}$ and K_2 (which come earlier in the multiple sum). *A priori*, $k_{1,1} \in \{0, \cdots, R\}$. However, $k_{1,1} = K_2 - k_{2,1} - k_{1,2} \leq K_2 - k_{2,1}$ and $k_{1,1} = K_2 - k_{2,1} - k_{1,2} \geq K_2 - k_{2,1} - R$, so we have $k_{1,1} \in \{\max\{0, K_2 - k_{2,1} - R\}, \cdots, \min\{R, K_2 - k_{2,1}\}\}$. These are the values of $k_{1,1}$ that are to be summed over, i.e. that are compatible with the values of previously specified indices. More generally,

$$\begin{aligned} k_{i,1} &= K_{i+1} - k_{i+1,1} - K_{i-1} - k_{i-1,2} - k_{i,2} \\ &\geq K_{i+1} - k_{i+1,1} - (2i-1)R \end{aligned}$$

$$\begin{aligned} k_{i,1} &= K_{i+1} - k_{i+1,1} - K_{i-1} - k_{i-1,2} - k_{i,2} \\ &\leq K_{i+1} - k_{i+1,1} \end{aligned}$$

$$0 \leq k_{i,1} \leq R$$

$$\begin{aligned} k_{i,1} \in \{&\max\{0, K_{i+1} - k_{i+1,1} - (2i-1)R\}, \cdots, \\ &\min\{R, K_{i+1} - k_{i+1,1}\}\} \end{aligned}$$

Similarly,

$$\begin{aligned} K_i &= K_{i+1} - k_{i+1,1} - k_{i,2} \\ &\geq K_{i+1} - k_{i+1,1} - R \end{aligned}$$

$$\begin{aligned} K_i &= K_{i+1} - k_{i+1,1} - k_{i,2} \\ &\leq K_{i+1} - k_{i+1,1} \end{aligned}$$

$$K_i \geq k_{i,1}$$

$$\begin{aligned} K_i \in \{&\max\{k_{i,1}, K_{i+1} - k_{i+1,1} - R\}, \cdots, \\ &K_{i+1} - k_{i+1,1}\} \end{aligned}$$

We redefine $b(k_{1:n}) = \mathrm{B}(k_{n-1,2} + k_{n,1}, K_{n-1})$ as the same quantity in terms of the new indices of summation, $b(K_{n-1}, K_n) \equiv \mathrm{B}(K_n - K_{n-1}, K_{n-1})$. Putting this all together we can write the normalization constant in the form of a sum-product algorithm, arranging the sums in the prescribed order and pushing all factors as far to the left as possible:

$$\begin{aligned} \mathcal{Z} = \sum_{K_T=0}^{R(2T-2)} &\left(\sum_{k_{T-1,1}=\max\{0, K_T-(2(T-2)-1)R\}}^{\min\{R, K_T\}} \right. \\ &\sum_{K_{T-1}=\max\{k_{T-1,1}, K_T - R\}}^{K_T} \\ &a_{[k_{T-1,1}, K_T - K_{T-1}]} b(K_{T-1}, K_T) \left(\cdots \left(\right.\right. \\ &\sum_{k_{3,1}=\max\{0, K_4 - k_{4,1} - 5R\}}^{\min\{R, K_4 - k_{4,1}\}} \sum_{K_3=\max\{k_{3,1}, K_4 - k_{4,1} - R\}}^{K_4 - k_{4,1}} \\ &a_{[k_{3,1}, K_4 - K_3 - k_{4,1}]} b(K_3, K_4) \left(\right. \\ &\sum_{k_{2,1}=\max\{0, K_3 - k_{3,1} - 3R\}}^{\min\{R, K_3 - k_{3,1}\}} \sum_{K_2=\max\{k_{2,1}, K_3 - k_{3,1} - R\}}^{K_3 - k_{3,1}} \\ &a_{[k_{2,1}, K_3 - K_2 - k_{3,1}]} b(K_2, K_3) \left(\right. \\ &\sum_{k_{1,1}=K_1=\max\{0, K_2 - k_{2,1} - R\}}^{\min\{R, K_2 - k_{2,1}\}} \\ &\left.\left.\left.\left.\left. a_{[k_{1,1}, K_2 - K_1 - k_{2,1}]} b(K_1, K_2) \right)\right)\right) \cdots \right)\right) \end{aligned} \tag{8}$$

Note that since we are no longer explicitly summing over $k_{i,2}$, it has been replaced by $K_{i+1} - K_i - k_{i+1,1}$

in the second element of the multi-index indexing the coefficients $\{a_k\}$. The forward variables $A^{(i,j)}$, then, can be immediately read off as follows. The first and second superscripts index values of $k_{t,1}$ and K_t, respectively:

$$A_2^{(i,j)} = \sum_{k=\max\{0,j-i-R\}}^{\min\{R,j-i\}} a_{[k,j-k-i]} \mathrm{B}(j-k,k)$$

$$i \in \{0,\cdots,R\}, j \in \{i,\cdots,i+2R\}$$

$$A_t^{(i,j)} = \sum_{k=\max\{0,j-i-(2t-3)R\}}^{\min\{R,j-i\}} \sum_{l=\max\{k,j-i-R\}}^{j-i} a_{[k,j-l-i]} \mathrm{B}(j-l,l) A_{t-1}^{(k,l)}$$

$$i \in \{0,\cdots,R\}, j \in \{i,\cdots,i+2(t-1)R\}$$

$$A_T^{(j)} = \sum_{k=\max\{0,j-(2t-4)R\}}^{\min\{R,j\}} \sum_{l=\max\{k,j-R\}}^{j} a_{[k,j-l]} \mathrm{B}(j-l,l) A_{T-1}^{(k,l)}$$

$$j \in \{0,\cdots,2(T-1)R\}$$

and $\mathcal{Z} = \sum_{i=0}^{2(T-1)R} A_T^{(i)}$. Similarly we can derive backward variables $C_t^{(i,j)}$, where the first superscript indexes $k_{t-1,2}$ and the second indexes $L_t \equiv \sum_{i=t}^{T} k_{i-1,2} + k_{i,1}$. Marginal quantities can be computed readily. The singleton marginal density is

$$p(x_t) = \frac{1}{\mathcal{Z}} \sum_{i=0}^{R} \sum_{j=i}^{i+2(t-2)R} \sum_{k=0}^{R} \sum_{l=k}^{k+2(T-1-t)R} A_{t-1}^{(i,j)} C_{t+1}^{(k,l)} \mathrm{B}(j,l) \sum_{k_{t-1,2}=0}^{R} \sum_{k_{t,1}=0}^{R} a_{[i,k_{t-1,2}]} a_{[k_{t,1},k]} x_t^{k_{t-1,2}+k_{t,1}} (1-x_t)^{j+l} \tag{9}$$

Therefore this method requires $O(R^2T^2)$ storage for the forward and backward variables, and $O(R^6T^2)$ processing time to compute marginals. For processing time, one factor of T comes from the number of marginals to compute. The other factor comes from the number of forward variables for each time, which increases linearly with T, manifesting in the limits of the sums over j and l in eq. (9).