
Low rank continuous-space graphical models

Carl Smith

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

Frank Wood and Liam Paninski

Department of Statistics
Columbia University
New York, NY 10027
{fwood, liam}@stats.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 graphical model class in which exact inference can be performed efficiently, owing to a certain “low-rank” structure in the potentials. While we focus on the case of tree graphical models, the low-rank treatment can also be applied for efficient exact inference in certain sparsely-loopy models. 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 arise when modeling survey responses conducted over many years in which the same yes/no question is asked but where 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 varies 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 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 data.

The aim of this work is to expand the class of models for which exact inference is computationally feasible. 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.

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.

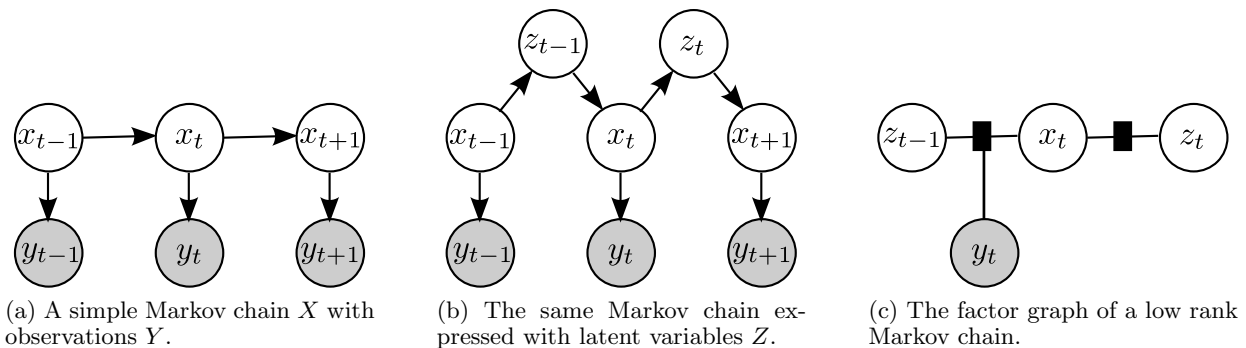


Figure 1

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 the graphical model in Figure 1 and consider the sequence of random variables $X = \{x_t\}_{t=1}^T$. 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 $z_1 \sim p(z_1|x_1)$, $x_2 \sim p(x_2|z_1)$, $z_2 \sim p(z_2|x_2)$, ..., then the marginal sequence x_1, \dots, 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, we can 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, by alternately sampling from $Z|X$, then $X|(Z, Y)$.

As the “number” of z samples is increased, neighboring values of x are more closely coupled together. The number of samples used in this scheme corresponds to a more general statement about the rank of a Markov random field potential linking neighboring x 's; a fact that we elaborate on in the following.

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. What has been overlooked until now (to our knowledge) is that the x 's can often be analytically marginalized out, leaving a Markov chain in z 's only, where computation remains tractable when the z 's are discrete random variables with a small state space. Thus, 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

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

We disregard the observations y_t for now. This can be re-expressed in the following equivalent form

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

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. (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) case, exact inference requires just $O(R^2)$ time (assuming constant $R_t = R$), as in a standard low-rank hidden Markov model; here the z 's correspond to the latent variables. Consider the partition function

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

We arrive at the distribution of Z simply by removing the sums over z_t . Rearranging the sums and integrals above reveals the Markov structure of Z .

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

We can use forward-backward 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, the forward and backward variables are as fol-

lows:

$$\begin{aligned} A_1^{(j_1)} &= \int dx_1 f_{1,j_1}(x_1) \\ A_t^{(j_t)} &= \sum_{j_{t-1}}^{R_{t-1}} A_{t-1}^{(j_{t-1})} \int dx_t g_{t-1,j_{t-1}}(x_t) f_{t,j_t}(x_t) \\ B_T^{(j_T)} &= \int dx_T g_{T-1,j_T}(x_T) \\ B_t^{(j_t)} &= \sum_{j_{t+1}}^{R_{t+1}} B_{t+1}^{(j_{t+1})} \int dx_t g_{t,j_t}(x_t) f_{t,j_{t+1}}(x_t) \quad (3) \end{aligned}$$

The validity of these expressions can be shown by induction on t . The marginal moments of X can be expressed in terms of the forward and backward variables:

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

To summarize, 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 $O(\sum_{t=1}^T R_t^2)$ time, by the forward-backward algorithm sketched above. (Note that we need only compute these inner products once; these can therefore be pre-tabulated 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, as discussed further below.

When employing such a model to model 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, one slight caveat: 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.

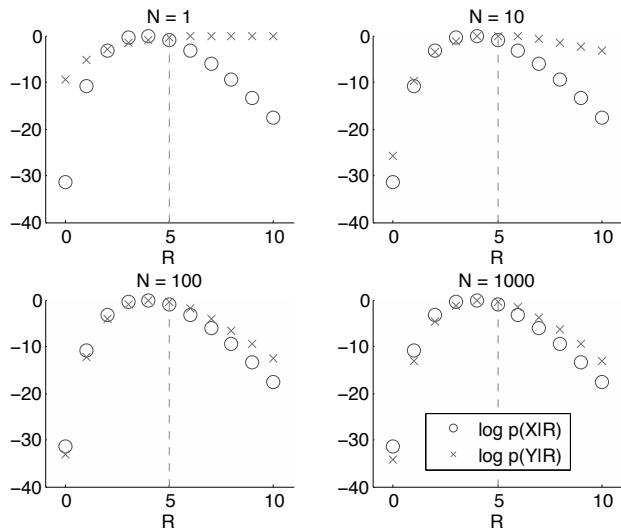


Figure 2: The marginal likelihood can be used to estimate the rank of the underlying process X generating data Y . Here a sample X_0 was generated from the beta-binomial time series model $p(X|R)$ with rank $R = 5$; i.e., $R_t = 5$ for all times t . We plot the loglikelihood $\log p(X_0|R)$ as a function of R . Then we generate data Y from $p(Y|X_0)$ and use the data to estimate the rank by maximizing the loglikelihood $\log p(Y|R)$ (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 $\text{Beta}(\alpha, \beta)$, 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} . Equation (1)

in this case becomes

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

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. Let

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

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 distribution; it is easy to see that the resulting smoothing prior $p(X)$ is conjugate to multinomial data, due

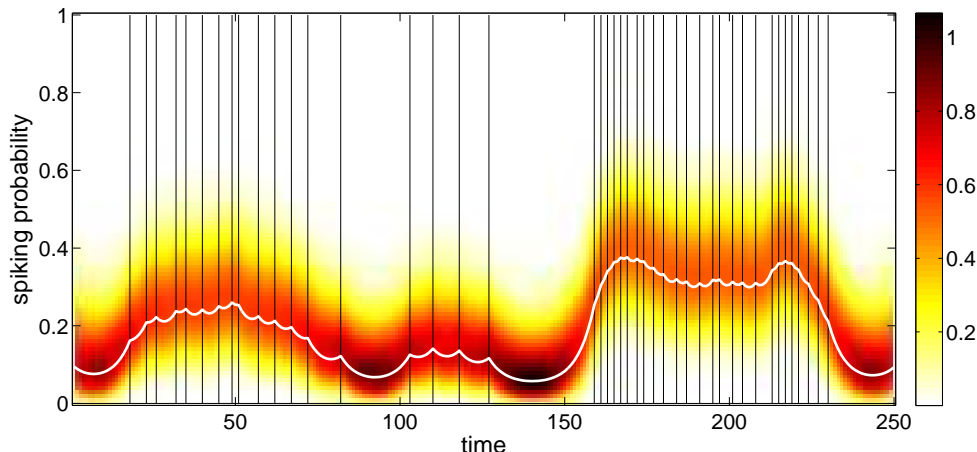


Figure 3: The inferred spiking probability density from spike train data assuming a binomial spiking model and the beta-binomial smoother. The black bars are the observed spikes. The solid white line is the inferred mean of the spiking probability. Each time unit is 2 ms.

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, $x_t \rightarrow q_t = (x_t \ s_t)$, where s_t denotes the cumulative sum $s_t = x_1 + x_2 + \dots + x_t$. This leads to a Markov prior on the augmented state variable Q of the form

$$p(Q) \propto \delta(x_1 = s_1) x_1^{\nu_1} \sum_{j_1} a_{j_1} x_1^{j_1} x_2^{j_2} \delta(x_2 = s_2 - s_1) \dots \delta(s_T = 1)$$

where $s_0 \equiv 0$. As outlined in greater detail in the appendix, we can perform forward backward inference on this density by recursively integrating the above density to compute the normalization constant, and then rearranging the summations into the form of a sum-product algorithm. The resulting inference algorithm requires $O(N^2T)$ storage and $O(N^6T)$ processing time.

4.3 Phase data

So far our random variables have lived in 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 arbitrary manifolds, 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

$$p(X) \propto \prod_t \sum_{i=0}^R e^{(R/2) \cos(x_t - \frac{2\pi i}{R+1})} e^{(R/2) \cos(x_{t+1} - \frac{2\pi i}{R+1})} \quad (5)$$

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 general-

ization). We will describe another generalization, to oscillatory or narrowband time series data, below.

5 Experiments

We began by analyzing some simple neural spike train data¹ 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 (4) described above 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 probability distribution over x_t as a function of time as shown in Figure 3. The results are qualitatively reasonable: the marginal mean varies smoothly over time, rising during times of higher spike rates.

We also performed some basic comparisons to Gibbs sampling in this model. 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 Database². Specifically, 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 (5) 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.

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

¹<http://neurotheory.columbia.edu/larry/book/exercises.html>

²<http://mocap.cs.cmu.edu/search.php?subjectnumber=13&trinum=9>

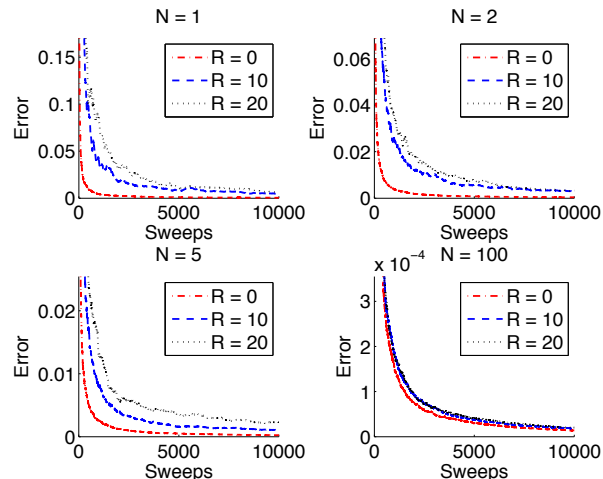


Figure 4: For different amounts N of data, the marginal means of a Markov chain X of length $T = 100$ were computed both exactly and approximately by Gibbs sampling, using the beta-binomial smoother. Here we plot the root mean square error per time step of the Gibbs solution with respect to the exact 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.

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 (Cadieu and Koepsell, 2010).

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”

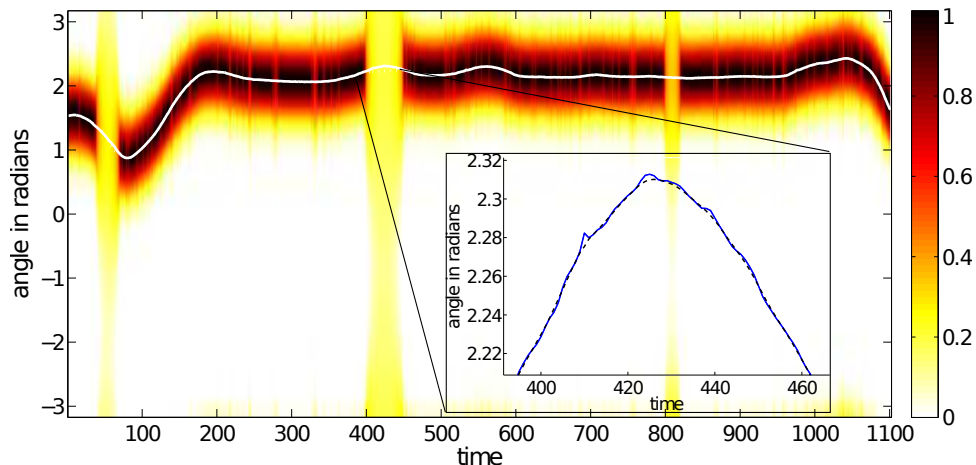


Figure 5: The inferred probability density of the angle in the motion capture data. The solid white line is the observed signal. The dotted white line, mostly obscured by the solid white line, is the inferred mean. Colorbar indicates the inferred posterior $p(x_t|Y)$. The yellow bands appear in the intervals where all the observations were suppressed. They are tapered because deeper within the band, where observations are farthest away, the density is relatively agnostic and therefore more nearly uniform. Inset: Data from inference with no data held out. The solid line is the observed signal. The dashed line is the inferred mean. Much of the evident noise in the signal has been smoothed. Each time unit is 2 ms.

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: investigating connections with recent work on inference via reproducing Hilbert space kernel methods by (Song et al., 2010b), (Song et al., 2010a); finding conditions in which tractable inference by belief propagation is possible in densely loopy graphs; and finding examples of such models with continuous z where efficient exact inference is possible.

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 drawn 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 should result in substantial improvements.

Appendix

In section 4.2 we describe a polynomial time smoother for multinomial data confined to the unit simplex. In this appendix we walk through a derivation of the forward variables of this smoother. We show that inference on the joint density is $O(N^2T)$ in storage and $O(N^6T^2)$ in processing time.

The joint density in the expanded state space is

$$\begin{aligned}
 p(Q) = & \sum_{k_1}^N a_{k_1} x_1^{\nu_1 + k_{11} - 1} \delta(x_1 = s_1 - s_0) \cdot \\
 & \sum_{k_2}^N a_{k_2} x_2^{\nu_2 + k_{12} + k_{21} - 1} \delta(x_2 = s_2 - s_1) \cdots \\
 & \sum_{k_{T-1}}^N a_{k_{T-1}} x_{T-1}^{\nu_{T-1} + k_{T-2,2} + k_{T-1,1}} \delta(x_{T-1} = s_{T-1} - s_{T-2}) \cdot \\
 & x_T^{\nu_T + k_{T-1,2} - 1} \delta(x_T = s_T - s_{T-1}) \delta(s_T = 1) \quad (6)
 \end{aligned}$$

where k_i is a multi-index $[k_{i1}, k_{i2}]$ and $s_0 \equiv 0$. The coefficients $\{a_i\}$ are chosen to be the same across time,

though this is only for notational convenience. Furthermore, for our purposes the $\{\nu_i - 1\}$ may be omitted and considered absorbed by the indices, to further simplify the notation. Lastly, we define $\delta_i \equiv \delta(x_i = s_i - s_{i-1})$. To perform inference, we are interested in recursively computing the normalization constant

$$\begin{aligned} Z &= \int_0^1 ds_1 \int_0^{s_1} dx_1 \cdots \int_0^1 ds_T \int_0^{s_T} dx_T p(Q) \\ &= \int_0^1 ds_1 \int_0^{s_1} dx_1 \sum_{k_1}^N a_{k_1} x_1^{k_{11}} \delta_1 \cdot \\ &\quad \int_0^1 ds_2 \int_0^{s_2} dx_2 \sum_{k_2}^N a_{k_2} x_2^{k_{12}+k_{21}} \delta_2 \cdots \end{aligned} \quad (7)$$

We integrate from left to right, first integratin dx_i , then ds_i .

$$\begin{aligned} Z &= \sum_{k_1}^N a_{k_1} \int_0^1 ds_1 (s_1 - s_0)^{k_{11}} \int_0^1 ds_2 \int_0^{s_2} dx_2 \\ &\quad \sum_{k_2} a_{k_2} x_2^{k_{12}+k_{21}} \delta_2 \cdots \\ &= \sum_{k_1} a_{k_1} \int_0^1 ds_2 \int_0^{s_2} dx_2 (s_2 - x_2)^{k_{11}} \cdot \\ &\quad \sum_{k_2} a_{k_2} x_2^{k_{12}+k_{21}} \cdots \\ &= \sum_{k_1} a_{k_1} \sum_{k_2} a_{k_2} B(k_{12} + k_{21}, k_{11}) \cdot \\ &\quad \int_0^1 ds_3 \int_0^{s_3} dx_3 (s_3 - x_3)^{k_{11}+k_{12}+k_{21}} \end{aligned} \quad (8)$$

where $B(\alpha, \beta)$ is the Beta function, and the last equality is found by a change of variables. Next define $k_{02} \equiv k_{T,1} \equiv 0$, and $K_i \equiv \sum_{j=1}^i k_{j-1,2} + k_{j,1}$. Lastly, define $b_i \equiv B(k_{i-1,2} + k_{i,1}, K_i)$. Continuing with the integration we soon find

$$\begin{aligned} Z &= \sum_{k_1}^N a_{k_1} \left(\sum_{k_2}^N a_{k_2} b_2 \left(\sum_{k_3}^N a_{k_3} b_3 \cdots \right. \right. \\ &\quad \left. \left. \left(\sum_{k_{T-2}}^N a_{k_{T-2}} b_{T-2} \left(\sum_{k_{T-1}}^N a_{k_{T-1}} b_{T-1} b_T \right) \right) \cdots \right) \right) \end{aligned} \quad (9)$$

Though this looks already to be in the form of a sum-product algorithm, it is not because B_i depends on the values of all the indices $k_{j \leq i}$. We can rearrange this summation into the form of a tractable sum-product

algorithm as follows.

$$\begin{aligned} Z &= \sum_{K_T=0}^{N(2T-2)} \left(\sum_{k_{n-1,1}=\max\{0, K_n - (2(T-2)-1)N\}}^{\min\{N, K_T\}} \cdot \right. \\ &\quad \left. a_{[k_{T-1,1}, K_T - K_{T-1}]} b_T \left(\cdots \left(\sum_{k_{11}=\max\{0, K_2 - k_{2,1} - N\}}^{\min\{N, K_2 - k_{2,1}\}} \right. \right. \right. \\ &\quad \left. \left. \left. a_{[k_{11}, K_2 - k_{11} - k_{21}]} b_2 \right) \cdots \right) \right) \end{aligned} \quad (10)$$

The forward variables $A^{(i,j)}$, then, are as follows.

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

$$i \in \{0, \dots, N\}, j \in \{i, \dots, i+2N\}$$

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

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

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

$$j \in \{0, \dots, 2(T-1)N\} \quad (11)$$

and $Z = \sum_{k=0}^{2T-2} A_T^{(k)}$. Similar expressions can be derived for backward variables $C_t^{(l,k)}$, and then marginal quantities can be computed readily. For instance, the singleton marginal density is

$$\begin{aligned} p(x_t) &= \frac{1}{Z} \sum_{i=0}^N \sum_{j=i}^{i+2(t-2)N} \sum_{k=0}^N \sum_{l=k}^{2(T-t-1)N} \\ &\quad A_{t-1}^{(i,j)} C_{t+1}^{(k,l)} B(j, l) \sum_{k_{t-1,2}=0}^N \sum_{k_{t,1}=0}^N \\ &\quad a_{[i, k_{t-1,2}]} a_{[k_{t,1}, k]} x_t^{k_{t-1,2}+k_{t,1}} (1-x_t)^{j+l} \end{aligned} \quad (12)$$

So we need $O(N^2T)$ storage for the forward and backward variables, and we need $O(N^6T^2)$ processing time to compute marginal quantities.

References

- Cadieu, C. 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 *23rd Conference on Uncertainty in Artificial Intelligence (UAI'2007)*, Vancouver, Canada, July 2007.
- Gasthaus, J., Wood, F., Görür, D., and Teh, Y. (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, Second Edition (Chapman & Hall/CRC Texts in Statistical Science)*. Chapman and Hall/CRC, 2 edition.
- Kass, R. and Raftery, A. (1995). Bayes factors. *Journal of the American Statistical Association*, 90:773–795.
- 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.
- 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*, 100:554–564.
- Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2).
- Siddiqi, S., Boots, B., and Gordon, G. (2010). Reduced-rank hidden Markov models. In *Proc. 13th Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*.
- Song, L., Gretton, A., and Guestrin, C. (2010a). Non-parametric tree graphical models via kernel embeddings. In *In Artificial Intelligence and Statistics (AISTATS)*.
- Song, L., Siddiqi, S. M., Gordon, G. J., and Smola, A. J. (2010b). Hilbert space embeddings of hidden markov models. In *International Conference on Machine Learning*, pages 991–998.
- Wainwright, M. and Jordan, M. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305.