Estimation and Selection for High-Order Markov Chains with Bayesian Mixture Transition Distribution Models

Matthew Heiner*

Department of Statistics, Brigham Young University

and

Athanasios Kottas†

Department of Statistics, University of California, Santa Cruz

July 15, 2020

Abstract

We develop a mixture model and diagnostic for Bayesian estimation and selection in high-order, discrete-state Markov chains. Both extend the mixture transition distribution, which constructs a transition probability tensor by aggregating probabilities from a set of single-lag transition matrices, through inclusion of mixture components dependent on multiple lags. We demonstrate two uses for the proposed model: identification of relevant lags through over-specification and shrinkage via priors for sparse probability vectors, and parsimonious approximation of multi-lag dynamics by mixing low-order transition models. The diagnostic yields a general and interpretable mixture decomposition for transition probability tensors estimated by any means. We demonstrate the utility of the model and diagnostic with simulation studies, and further apply the methodology to a data analysis from the high-order Markov chain literature, and to a time series of pink salmon abundance in Alaska, U.S.A.

Keywords: Categorical time series, Dimension reduction, Model selection, Shrinkage prior, Tensor decomposition

*M. Heiner (heiner@stat.byu.edu) is Assistant Professor, Department of Statistics, Brigham Young University.
†A. Kottas (thanos@soe.ucsc.edu) is Professor, Department of Statistics, University of California, Santa Cruz.
1 Introduction

Consider modeling a time series of nominal or ordinal values $s_t \in \{1, \ldots, K\}$ collected at equally spaced, discrete times $t = 1, \ldots, T$. A popular approach for capturing serial correlation is to assume Markovian dynamics: that the conditional probability distribution of $s_t$ depends only on the recent past. Time homogeneity, or time invariance of the transition probabilities, is also typically assumed. These simplifying assumptions, nearly essential for inference in small or moderate sample size scenarios, are often appropriate even if the time series is not truly Markovian. Another common assumption is to condition only on the single most recent lag. However, restricting a model to first-order dynamics, or even selecting the incorrect lag, can miss important features in the data. In this article, we propose Bayesian models to address two distinct objectives: identification of the relevant time-delay coordinates, with the effective Markovian order; and parsimonious modeling of high-order chains.

Assuming time homogeneity, a full, unrestricted single-lag model requires estimation of $K$ discrete distributions, each with $K - 1$ free parameters. A Markov chain dependent on the first $L$ lags, traditionally called a chain of order $L$, requires estimation of $K^L$ such distributions, limiting consideration to low orders for most time series. Typically, model order is selected by maximizing a (possibly penalized) likelihood (Katz, 1981; Raftery, 1985; Prado and West, 2010), performing trans-dimensional MCMC (Green, 1995; Insua et al., 2012), using Bayes factors (Fan and Tsai, 1999; Bacallado, 2011; Zucchini and MacDonald, 2009), predictive/information criteria, or goodness-of-fit tests (Bartlett, 1951; Besag and Mondal, 2013). Each of these approaches requires either fitting multiple models or complex estimation methods. Our approach is to build lag inference into a single model.

Several approaches have been proposed to address exponential growth in the parameter space for higher-order transitions. Raftery (1985) introduced the mixture transition distribution (MTD), a general-purpose, parsimonious model for Markov chains. The MTD model was extended in Raftery and Tavaré (1994) and further developed in the decade following; see Berchtold and Raftery (2002) for a review. In the original MTD model, lags contribute to the transition probabilities by mixing over a single transition matrix. Only
one new parameter is added for each additional lag.

Contemporary with the MTD model, generalized linear models for multinomial outcomes were applied to categorical time series (Liang and Zeger, 1986; Zeger and Liang, 1986; Fahrmeir and Kaufmann, 1987). These models can accommodate varying degrees of complexity by controlling the order of interactions among the linear predictors (lags), up to and including a full model order with \( K^L(K - 1) \) parameters. These models can also account for exogenous sources of non-stationarity through covariates. However, estimation and interpretability become problematic in these models when many lags are considered.

Tree-based methods provide an alternative parsimonious approach. Variable-length Markov chains (VLMC, Ron et al., 1994; Bühlmann et al., 1999) reduce the parameter space by clustering the \( K^L \) transition distributions via recursive pruning. Sparse Markov chains (SMC, Jääskinen et al., 2014) partition the \( L \)-dimensional lag space without hierarchical constraints, resulting in greater flexibility. They also feature a prior structure which encourages low orders. Although efficient, these models lack posterior uncertainty quantification, and inferences for order and lag relevance are not readily available.

More recently, Sarkar and Dunson (2016) proposed a Bayesian nonparametric model for high-order Markov chains. They model the \( K^L \) transition distributions through tensor factorization and further encourage parsimony by clustering the components of a core mixing distribution with a Dirichlet process prior (Ferguson, 1973). By allowing variable dimensions along different modes of the core mixing distribution, the model further admits inferences for lag inclusion. This model enjoys a fully Bayesian, albeit complicated, implementation and performs well against the methods described above in forecasting in scenarios with up to four states and ten lags.

Our modeling strategy is to build on the simplicity and interpretability of the MTD model. Our proposed extension includes mixture components that depend on more than one lag, together with inference for the set of relevant lags. As a byproduct, the model estimates the effective order, or number of relevant lags on which the transition depends, which may be lower than the model order. We employ a Bayesian framework with conditionally conjugate priors that encourage sparsity and parsimony in the MTD structure. We further
introduce a model diagnostic procedure that can be applied as a general and interpretable
decomposition of estimated transition probability tensors.

The remainder of this paper is organized as follows. In Section 2, we review the MTD
model and develop our proposed extension, along with our approach for Bayesian infer-
ence using structured priors to aid with the model’s intended uses. Section 3 introduces
the high-order MTD decomposition, together with an algorithm for its computation. In
Section 4, we test the proposed model using two simulation scenarios that reflect our two
objectives, demonstrating improved predictive performance over the original MTD. Section
5 illustrates the models through two analyses, first on a data set which appears in the pre-
ceding literature, and second on annual time series of pink salmon abundance in Alaska,
U.S.A. Finally, we conclude with a discussion in Section 6. Technical details are provided
in the appendix.

2 Modeling approach

In a full \(L\)-order, time-homogeneous Markov chain, the collection of all possible transition
probabilities \(\Pr(S_t = k_0 \mid S_{t-1} = k_1, \ldots, S_{t-L} = k_L)\), for \(k_\ell \in \{1, \ldots, K\}, \ell \in \{1, \ldots, L\}\),
can be arranged in a \((L+1)\)-mode transition probability tensor with entries \((\Omega)_{k_0,k_1,\ldots,k_L} \in [0,1]\)
that sum to one along the first mode. If we condition on the first \(L\) observations of a
time series \(\{s_t: t = 1, \ldots, T\}\), the joint sampling distribution for the remaining sequence
is given by \(\Pr(\{s_t\}_{t=L+1}^T \mid \{s_t\}_{t=1}^L, \Omega) = \prod_{t=L+1}^T (\Omega)_{s_t,s_{t-1},\ldots,s_{t-L}},\) defining the conditional
likelihood that we employ hereafter.

We begin by specifying the original MTD model in Section 2.1 and motivate its exten-
sion. In Section 2.2, we propose a mixture model with additional levels of higher-order
interactions. The Bayesian implementation in Section 2.4 utilizes priors for sparse proba-
bility vectors discussed in Section 2.3.
2.1 Original mixture transition distribution

The mixture transition distribution model constructs the transition probability tensor $\Omega$ as linear combinations of probabilities from a single column-stochastic matrix $Q$ and adds one parameter for each additional lag ($\lambda_\ell$), similar to autoregressive models. The transition probabilities in a model of order $L$ are given as

$$\Pr(S_t = k_0 \mid S_{t-1} = k_1, \ldots, S_{t-L} = k_L) = \sum_{\ell=1}^{L} \lambda_\ell (Q)_{k_0, k_\ell},$$

where $0 \leq \lambda_\ell \leq 1$ and $\sum_{\ell=1}^{L} \lambda_\ell = 1$. Although this model incorporates information beyond the first lag, it is restrictive in that it cannot capture nonlinear (non-additive) dynamics in more than one dimension of the lag space.

Form (1) suggests that lags with a prominent role in the transition probability will have relatively large $\lambda_\ell$ values. Indeed, inferences for $\lambda = (\lambda_1, \ldots, \lambda_L)$ yield information about lag relevance for the Markov process. It is apparent from (1) that $\lambda_\ell = 0$ is sufficient for conditional independence of $S_t$ and $S_{t-\ell}$ in the model. If the columns of $Q$ are distinct, then $\lambda_\ell = 0$ is also a necessary condition for conditional independence.

Inferences on $\lambda$ have facilitated at least informal assessment of lag relevance, as in Raftery and Tavaré (1994), although information criteria constitute the standard method for assessing order (Berchtold and Raftery, 2002). Recently, Tank et al. (2017) employed a likelihood penalty on $\lambda$ in a modified MTD model to infer Granger causality in the context of multiple time series. Heiner et al. (2019) implemented a Bayesian MTD model, utilizing shrinkage priors on $\lambda$ for insight into lag relevance. We adopt a similar approach here.

2.2 Mixtures of higher-order MTD components

MTD models offer parsimonious and interpretable representations for Markov chains with dependence extending beyond the most recent lag. Mixing over single-lag transitions, however, is quite restrictive. In their survey of generalizations for the MTD, Berchtold and Raftery (2002) suggest, but do not pursue, the possibility of mixing over higher-order transition tensors. We build a Bayesian framework for such an extension to include
higher-order “interactions,” and we refer to the resulting model as the mixture of mixture transition distributions (MMTD) model.

To define the MMTD model, let $R \leq L$ be a positive integer representing highest number of active lags in sub-models over which we mix. Typically, $R$ is small relative to $L$. The model comprises $R + 1$ levels. At the zeroth, or intercept level, we have $Q^{(0)}$, a length-$K$ probability vector; the first level uses $Q^{(1)}$, a $K \times K$ transition probability matrix; the second uses $Q^{(2)}$, a $K^3$ transition probability tensor; and so forth up to $Q^{(R)}$, a $K^{R+1}$ transition probability tensor, such that $\sum_{k_1=1}^{K}(Q^{(R)})_{k_1,\ldots,k_R} = 1$ for all $(k_1, k_2, \ldots, k_R) \in \{1, \ldots, K\}^R$. Next, introduce a mixing probability vector across levels, $\Lambda = (\Lambda_0, \Lambda_1, \ldots, \Lambda_R)$. The MMTD model for transition probabilities is given by

$$
\Pr(\mathcal{S}_t = k_0 | \mathcal{S}_{t-1} = k_1, \ldots, \mathcal{S}_{t-L} = k_L) = (\Omega)_{k_0, k_1, \ldots, k_L} = \Lambda_0 (Q^{(0)})_{k_0} + \Lambda_1 \sum_{\ell=1}^{L} \lambda^{(1)}_{\ell} (Q^{(1)})_{k_0, k_\ell} + \Lambda_2 \sum_{1\leq \ell_1 < \ell_2 \leq L} \lambda^{(2)}_{(\ell_1, \ell_2)} (Q^{(2)})_{k_0, k_{\ell_1}, k_{\ell_2}} + \ldots + \Lambda_R \sum_{1\leq \ell_1 < \ldots < \ell_R \leq L} \lambda^{(R)}_{(\ell_1, \ldots, \ell_R)} (Q^{(R)})_{k_0, k_{\ell_1}, \ldots, k_{\ell_R}},
$$

where $\lambda^{(r)}$ is a probability vector of length $\binom{L}{r}$, for $r = 1, \ldots, R$. This mixture of mixtures is equivalent to using a single (albeit long) $\lambda$ probability vector to mix over all possible arrangements of lags and base transition tensors $Q^{(r)}$. However, this parameterization is more interpretable when considering interaction-level relevance, indicated by $\Lambda$, as opposed to selection of lag configuration, indicated by $\lambda^{(r)}$. If $\Lambda_1 = 1$, we recover the original MTD model. The fully-parameterized transitions associated with $Q^{(r)}$ allow unrestricted dynamics in $r$ dimensions of the lag space. As a discrete probability mixture of probability distributions, this model produces a valid conditional probability distribution.

The model in (2) is clearly over-parameterized. For example, it is possible for a higher-level $Q^{(r)}$ to mimic a lower-level tensor through repetition of transition probabilities across values of an inactive lag. Consequently, $\Lambda$, $\{\lambda^{(r)}\}_{r=1}^{R}$, and $\{Q^{(r)}\}_{r=0}^{R}$ lack identifiability, complicating their use for inference on the dependence structure. Tank et al. (2017) address
a similar issue for MTD models by reparameterizing to include a zeroth (intercept) level, and shrinking estimation toward a unique, identifiable representation, wherein a maximal amount of transition probability mass is transferred to the distribution in the intercept. In Section 3, we extend this concept to mixtures of higher-order transition tensors. In the interim, we propose a Bayesian model that, while not fully identified, shrinks the mixture weights to favor lower levels of the model.

We envision two primary uses for the MMTD model. The first is to uncover a dependence substructure from time series whose effective order is less than or equal to a pre-specified $R$, in which case the true model is contained within the mixture framework, provided $L$ is high enough. For example, we might postulate that a time series has effective order two, but we are unsure which two lags are important. Assuming a maximal horizon of five lags, we could fit the MMTD model with $L = 5$ and $R = 2$, or even $R = 3$. If the effective order is indeed two, we would anticipate that $\Lambda_2$ would carry substantial posterior weight, and that inferences for $\lambda^{(2)}$ would identify the influential lags.

If the true effective order of dependence in the time series is greater than $R$, our second intended use for the MMTD model is analogous to that of the MTD, wherein we parsimoniously approximate higher-order dependence through a mixture of lower-order transition tensors. Using $R > 1$ may be thought of as including interaction-like terms in the mixture.

Our proposed model formulation requires estimation of $R$ free $\Lambda$ parameters, $\binom{L}{1} + \binom{L}{2} + \ldots + \binom{L}{R} - R$ free $\lambda$ parameters, and $(K - 1) + K(K - 1) + K^2(K - 1) + \ldots + K^R(K - 1) = K^{R+1} - 1$ free parameters in $\{Q^{(r)}\}_{r=0}^R$. The fastest-growing term in the $\lambda$ parameter count increases no faster than a polynomial in $L$ of degree $\lceil R/2 \rceil$ divided by $R!$, while the transition distributions grow exponentially. Table 1 reports free parameter counts for different combinations of $K$, $L$, and $R$. Typically, $K$ is fixed and known, and a modeler must select $L$ and $R$ considering parsimony, estimability for a given sample size, and computational cost. If $R$ is much smaller than $L$, the MMTD substantially reduces the parameter space from the original full-order Markov chain. The parameter space is effectively further reduced by shrinkage priors on $\Lambda$ and $\{\lambda^{(r)}\}$. 

7
Table 1: Free parameter count for MMTD model under different combinations of state-space size $K$, largest possible lag $L$, and largest mixing order $R$. The total number of parameters is the sum of the free $\Lambda$, $\lambda$, and $Q$ parameters. The unrestricted total is the number of parameters required to estimate an unrestricted transition probability tensor of order $L$.

<table>
<thead>
<tr>
<th>$K$</th>
<th>$L$</th>
<th>$R$</th>
<th>$\Lambda$</th>
<th>$\lambda$</th>
<th>$Q$</th>
<th>Total</th>
<th>Unrestricted</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>13</td>
<td>7</td>
<td>22</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>26</td>
<td>31</td>
<td>61</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>53</td>
<td>7</td>
<td>62</td>
<td>1,024</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>4</td>
<td>4</td>
<td>381</td>
<td>31</td>
<td>416</td>
<td>1,024</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>13</td>
<td>124</td>
<td>139</td>
<td>12,500</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>26</td>
<td>3,124</td>
<td>3,154</td>
<td>12,500</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>53</td>
<td>124</td>
<td>179</td>
<td>$3.91 \times 10^7$</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>4</td>
<td>4</td>
<td>381</td>
<td>3,124</td>
<td>3,509</td>
<td>$3.91 \times 10^7$</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>13</td>
<td>342</td>
<td>357</td>
<td>100,842</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>53</td>
<td>342</td>
<td>397</td>
<td>$1.69 \times 10^9$</td>
</tr>
</tbody>
</table>

2.3 Bayesian model with shrinkage priors

We present a Bayesian modeling approach for the MMTD, employing carefully chosen prior distributions that favor lower levels of interaction and encourage identification of low-order substructure. Specifically, we use two prior distributions for probability vectors, introduced by Heiner et al. (2019), that go beyond the standard Dirichlet prior by enforcing sparsity in the presence of data, as well as conditional stochastic ordering. Both prior distributions are continuous, bypassing problems that arise from the sum-to-one constraint when using priors with point masses.

The first is the sparse Dirichlet mixture (SDM) prior. This one-parameter extension of the Dirichlet distribution is a fixed-weight mixture of Dirichlet distributions, with each component featuring a boost of equivalent sample size $\beta > 1$ in one of the categories. If $\theta$ is a probability vector of length $J$, the SDM density is given as

$$p_{\text{SDM}}(\theta; \alpha, \beta) = \sum_{j=1}^{J} \frac{w_j}{\sum_{i=1}^{J} w_i} \text{Dir}(\theta; \alpha + \beta e_j),$$

where $\text{Dir}(\theta; \alpha + \beta e_j)$ denotes the Dirichlet density with shape parameter vector $\alpha + \beta e_j$, with $\alpha = (\alpha_1, \ldots, \alpha_J)$, where $e_j$ is a vector of zeros with a one in the $j$th position, and
\( w_j \equiv \prod_{h=1}^{J} \Gamma(\alpha_{h} + \beta_{1(h=j)}) \) with \( \Gamma(\cdot) \) denoting the gamma function. For small sample sizes and relatively large \( \beta \), the SDM can be described as a winner-takes-all prior in that it shifts most mass toward the \( \theta_j = 1 \) corner of the supporting simplex for the component with largest \( \alpha_j \).

The second prior is the stick-breaking mixture (SBM) prior. It builds the probability vector \( \theta \) through an extension of the stick-breaking construction that defines the generalized Dirichlet distribution (Connor and Mosimann, 1969). In particular,

\[
\theta_1 = X_1, \quad \theta_j = X_j \prod_{i=1}^{j-1} (1 - X_i) \text{ for } j = 2, \ldots, J - 1, \text{ and } \theta_J = \prod_{i=1}^{J-1} (1 - X_i),
\]

with \( X_j \) independently drawn from a mixture of three beta distributions, \( X_j \sim \pi_1 \text{Beta}(1, \eta) + \pi_2 \text{Beta}(\gamma_j, \delta_j) + \pi_3 \text{Beta}(\eta, 1) \), where \( \pi_1 + \pi_2 + \pi_3 = 1 \). This mixture structure encourages sparsity when \( \eta \) is large, in which case the first component corresponds to small probabilities in \( \theta \). The third component allows for the rest of the unbroken stick (i.e., \( \prod_{i=1}^{j-1} (1 - X_i) = 1 - \sum_{i=1}^{j-1} \theta_i \)) to be used for \( \theta_j \), while the second mixture component allows for flexibility in modeling \( \theta_j \). The \( \gamma_j \) and \( \delta_j \) parameters can be fixed at the same values across \( j \), or can be set to mimic the Dirichlet distribution as with the generalized Dirichlet distribution, resulting in a three-parameter extension of the Dirichlet distribution. To accommodate the MMTD model, we adapt the SBM prior to allow the mixture weights \( \pi_1, \pi_2, \) and \( \pi_3 \) to vary with \( j \).

Both the SDM and SBM priors enjoy conditional conjugacy with multinomial sampling models. If the hyperparameters of the priors are fixed, as is common with Dirichlet priors, incorporating them into a hierarchical model yields direct Gibbs sampling and admits swapping priors without structural changes to the updates outlined in Appendix A.

To build intuition on the role of the two priors, consider an example in which a time series exhibits an effective order of two. If the conditional distributions cannot be approximated as a mixture of single-lag transition probabilities, we anticipate that \( \Lambda_2 \) would carry substantial posterior weight, and that inferences for \( \lambda^{(2)} \) would identify the two influential lags. In this model-selection scenario, the SBM (on \( \Lambda \)) and SDM (on \( \lambda \)) priors contribute
In light of the model structure (2), model objectives, and identifiability considerations in Section 2.2, we carefully specify the SBM prior for the level weights in $\Lambda$. The stick-breaking mechanism in the prior provides (quasi) stochastic ordering with jumps, encouraging concentration at lower levels of interaction. Specifically, the intercept weight is drawn first, and with no penalty, that is, $\Lambda_0 \sim \text{Beta}(\gamma_0, \delta_0)$. We use either $\gamma_0 = \delta_0 = 1$ (corresponding to the uniform distribution) or beta shape parameters that favor large values of $\Lambda_0$. The rest of $\Lambda$ is broken sequentially from what remains in the unbroken stick. The remaining beta mixtures use $\pi_1 > 0$ and $\pi_3 > 0$ to regularize $\{\Lambda_r\}_{r=1}^R$. Setting $\pi_1 > 0$ allows for small values of the corresponding $\Lambda_r$, effectively skipping the $r$th level. Setting $\pi_3 > 0$ promotes consumption of the remaining mass before reaching $\Lambda_R$.

If a modeler employs the MMTD to discover an active sub-model, we advocate using the SDM prior on each $\lambda^{(r)}$ with a large value of $\beta$ to concentrate mass and select the best lag configuration. This use is further facilitated by the MMTD model having only one $Q^{(r)}$ at each level.

A modeler electing to use the MMTD as a true mixture (the second intended use) may still employ the SDM prior for each $\lambda^{(r)}$, but with a lower value of $\beta$ to encourage more mixing (note that $\beta = 1$ yields a Dirichlet prior). Note that the SBM prior on $\Lambda$ further allows mixing across orders, so that different levels of the model may mix across non-overlapping sets of lag configurations, potentially yielding an effective order higher than the highest selected level, or even higher than $R$.

The model is completed with independent priors for each of the $K^r$ probability distributions in $Q^{(r)}$, for $r = 0, 1, \ldots, R$. The traditional choice for transition matrices, or probability vectors in general, is to use Dirichlet distributions, which we adopt here. In cases of nearly deterministic dynamics, Heiner et al. (2019) found advantage in using independent SBM priors for each column of $Q$ in the standard MTD model. However, the MMTD model can spread estimation thin across multiple $Q$ tensors, relying more heavily on priors. This can potentially introduce undesired artifacts in the estimated transition probabilities, as the SBM prior is not symmetric. Results in Tank et al. (2017) and the pro-
procedure introduced in Section 3 suggest that increasing the variance of component transition distributions in the MMTD model could further promote transferring weight to lower levels. While the SDM prior promotes this objective, preliminary results show no substantial gain, and we do not pursue its use further.

2.4 Inference and computation

To obtain full posterior inference under the MMTD model, we utilize a Gibbs sampler that alternates between collapsed conditional distributions made tractable by augmentation with latent configuration variables (Insua et al., 2012, pp. 59-60). As noted earlier, all inferences condition on the first $L$ observations in the time series $\{s_t\}_{t=1}^T$.

The sampling distribution for the time series is given as the product of transition probabilities in (2) across $t = L + 1, \ldots, T$. We break the mixture in (2) by introducing latent configuration variables $Z_t$ such that $\Pr(Z_t = r \mid \Lambda) = \Lambda_r$, for $r = 0, 1, \ldots, R$, independently for each observation time. Then conditional on $Z_t$ (and for $Z_t > 0$), further introduce $z_t$ such that $\Pr(z_t = (\ell_1, \ldots, \ell_r) \mid Z_t = r, \lambda^{(r)}) = \lambda^{(r)}_{(\ell_1, \ldots, \ell_r)}$, for $1 \leq \ell_1 < \ldots < \ell_r \leq L$, independently, for each observation time. The hierarchical formulation for this model is given in generative order as follows. For $t = L + 1, \ldots, T$; $k = 1, \ldots, K$; $k_\ell = 1, \ldots, K$; $\ell = 1, \ldots, L$, $1 \leq \ell_1 < \ldots < \ell_r \leq L$; and $r = 0, 1, \ldots, R$, we have

$$Q^{(0)} \sim \text{Dir}(\alpha_{Q^{(0)}}), \quad (Q^{(r)}),_{k_1, \ldots, k_r} \overset{\text{ind.}}{\sim} \text{Dir}(\alpha_{Q^{(r)}}), \quad \text{for } (k_1, \ldots, k_r) \in \{1, \ldots, K\}^r,$$

$$\Lambda \sim \text{SBM}(\pi_1, \pi_3, \eta, \gamma, \delta), \quad \lambda^{(r)} \overset{\text{ind.}}{\sim} \text{SDM}(\alpha_{\lambda^{(r)}}, \beta_{\lambda^{(r)}}), \quad \text{for } r = 1, \ldots, R,$$

$$\Pr(Z_t = r \mid \Lambda) = \Lambda_r, \quad \Pr(z_t = (\ell_1, \ldots, \ell_r) \mid Z_t = r, \lambda^{(r)}) = \lambda^{(r)}_{(\ell_1, \ldots, \ell_r)}, \quad \text{(5)}$$

$$\Pr(s_t = k \mid s_{t-1} = k_1, \ldots, s_{t-L} = k_L, Z_t = r, z_t = (\ell_1, \ldots, \ell_r), Q^{(r)}) = (Q^{(r)})_{k, k_\ell_1, \ldots, k_\ell_r},$$

where $\alpha_Q$ is a length-$K$ vector of positive shape parameters (which could potentially be separately specified for each distribution in each $Q$); $\pi_1$ and $\pi_3$ are length-$R$ vectors containing probabilities such that $(\pi_1)_r + (\pi_3)_r < 1$; $\gamma$ and $\delta$ are length-$R$ vectors containing positive shape parameters; $\alpha_{\lambda^{(r)}}$ is a length-$\binom{L}{r}$ vector of positive shape parameters; and
\( \beta_{\lambda(r)} > 1 \) is the SDM sparsity parameter. We always set \((\pi_1)_0 = 0\) to avoid penalizing \(\Lambda_0\), and recommend setting \((\pi_3)_0 = 0\) as well. Note that all quantities in (5) without explicit dependence are considered independent a priori.

This structure admits closed-form conditional sampling. To simplify computation, we uniquely map all \(Z_t\) and \(z_t\) pairs onto a single variable \(\zeta_t \in \{0, 1, \ldots, [(L_1^r) + (L_2^r) + \ldots + (L_R^r)]\}\) whose prior probability under the model is equal to the product of the corresponding \(\Lambda\) and \(\lambda\). Full conditional distributions for \(\Lambda\), each \(\lambda(r)\), and each probability vector in \{\(Q(r)\)\} are exactly analogous to multinomial-SBM, multinomial-SDM, and multinomial-Dirichlet conjugate updates, respectively, where \(Z_t\), \(z_t\), and observed data transitions supply the respective multinomial counts. Updates for \(Z_t\) and \(z_t\) (equivalently \(\zeta_t\)) require calculation of a discrete distribution. Details are given in Appendix A.

Standard Gibbs sampling is ineffective due to the model’s lack of identifiability. We therefore integrate \{\(Q(r)\)\}_{r=0}^{R} from the joint posterior, and sample the collapsed conditional distribution for \{\(\zeta_t\)\}. To encourage occasional jumps between modes of the posterior, we also include a hybrid independence-Metropolis step which jointly proposes \(\Lambda\), each \(\lambda(r)\), and \{\(\zeta_t\)\} from their joint prior every 5 or 10 iterations of the MCMC algorithm.

The augmented Gibbs sampler becomes computationally demanding as \(R\) and \(L\) increase because updates for the latent configuration variables \{\(\zeta_t\)\} involve calculation of \(\sum_{r=0}^{R} (L_r)\) probabilities for each time point \(t = L + 1, \ldots, T\). MCMC samplers for \(\Lambda\), \{\(\lambda\)\}, and \{\(Q\)\} utilizing the mixture likelihood based on (2) may provide an alternate strategy if \(K\) is reasonably small. Moving to a purely continuous parameter space does not, however, eliminate multimodality from the posterior distribution.

We conduct posterior sampling using the \textit{Julia} scientific computing language (Bezanson et al., 2017). Table 2 reports running time, in minutes per 1,000 MCMC iterations, for a selection of MMTD models fit in Sections 4 and 5. Reported times are averaged from two independent runs of 100,000 iterations after burn-in. The ratio of iterations per effective sample of log-likelihood ranges from single digits to approximately 1,000, and depends on the data and model settings. For example, clear identification of an active lag configuration as a subset of the model specification can increase efficiency by an order of magnitude.
Table 2: Running time in minutes per 1,000 MCMC iterations for the MMTD model at sample sizes and settings used in the simulation studies and data illustrations.

<table>
<thead>
<tr>
<th>K</th>
<th>L</th>
<th>R</th>
<th>T</th>
<th>Time</th>
<th>K</th>
<th>L</th>
<th>R</th>
<th>T</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>7</td>
<td>2</td>
<td>100</td>
<td>0.171</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>200</td>
<td>0.129</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>2</td>
<td>200</td>
<td>0.365</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>500</td>
<td>0.241</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>2</td>
<td>500</td>
<td>0.877</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>200</td>
<td>0.567</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>7</td>
<td>100</td>
<td>1.883</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>500</td>
<td>1.449</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>7</td>
<td>200</td>
<td>3.871</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>200</td>
<td>1.546</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>7</td>
<td>500</td>
<td>9.284</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>500</td>
<td>3.811</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>4</td>
<td>204</td>
<td>10.512</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>30</td>
<td>0.035</td>
</tr>
</tbody>
</table>

3 Mixture decomposition of transition probability tensors

In this section, we extend the MTD reduction technique of Tank et al. (2017) to transition probability tensors of arbitrary order. The resulting decomposition helps with identifiability of MMTD models, which the priors in Section 2.3 encourage, but do not enforce. Absent constraints, it is possible for lower-order patterns to embed in higher-order transition tensors, potentially causing MMTD model fits to artificially report dependence on more lags than necessary. Furthermore, this behavior is difficult to detect in high dimensions. To address these difficulties, we introduce a procedure that 1) reveals low-order embeddings, and 2) yields a unique and interpretable mixture decomposition. The decomposition is not itself a viable statistical model, but rather provides a diagnostic tool. We first illustrate the procedure by reducing a second-order transition tensor. We then outline the general algorithm and discuss its potential uses.

3.1 Illustration

Consider the transition probabilities in Table 3 for a binary chain whose transition probabilities depend on the first two lags. These probabilities could be organized into a $2^3$ transition tensor, or flattened into a matrix resembling Table 3.

The first step of reduction is to extract the intercept, i.e., the probabilities common to all four transition distributions. This is accomplished by removing the minima along rows
Table 3: Example transition probability distribution for a second-order binary chain.

<table>
<thead>
<tr>
<th></th>
<th>$S_{t-2}$ value</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{t-1}$ value</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

$$
\text{Pr}(S_t = 1 \mid S_{t-1}, S_{t-2}) = \begin{pmatrix} .5 & .4 & .9 & .2 \end{pmatrix}, \quad \text{Pr}(S_t = 2 \mid S_{t-1}, S_{t-2}) = \begin{pmatrix} .5 & .6 & .1 & .8 \end{pmatrix}.
$$

of Table 3 and collecting them in a length-$K$ vector $m$. We collect the “leftover” transition mass in $B^{(0)}$. Thus the first step yields

$$
m = \begin{pmatrix} .2 \\ .1 \end{pmatrix}, \quad \text{and } B^{(0)} = \begin{pmatrix} .3 & .2 & .7 & 0 \\ .4 & .5 & 0 & .7 \end{pmatrix}.
$$

The next step is to sequentially extract single-lag probability distributions. To identify the most influential single lag, we collect sets of transition matrices, with each set associated with a specific lag. The matrices associated with a given lag are obtained by fixing the other lags at their various values. They are, for lags 1 and 2 respectively,

$$
\begin{align*}
\text{lag 1: } & \begin{pmatrix} .3 & .2 \\ .4 & .5 \end{pmatrix} \text{ for } S_{t-2} = 1, \quad \begin{pmatrix} .7 & 0 \\ 0 & .7 \end{pmatrix} \text{ for } S_{t-2} = 2; \\
\text{lag 2: } & \begin{pmatrix} .3 & .7 \\ .4 & 0 \end{pmatrix} \text{ for } S_{t-1} = 1, \quad \begin{pmatrix} .2 & 0 \\ .5 & .7 \end{pmatrix} \text{ for } S_{t-1} = 2.
\end{align*}
$$

Our objective is to select one of the lags and derive from its associated matrices a common matrix to extract from each. Because we can extract no more than the minimal value at each matrix position in a given set, we collect the minima across matrices at each entry,

$$
\begin{align*}
\text{lag 1: } & M_1 = \begin{pmatrix} .3 & 0 \\ 0 & .5 \end{pmatrix}; \\
\text{lag 2: } & M_2 = \begin{pmatrix} .2 & 0 \\ .4 & 0 \end{pmatrix}.
\end{align*}
$$

The matrices in (8) provide a basis for selecting the first single-lag transition mass matrix, denoted $A^{(1)}$. The superscript indicates the number of lags in the extracted transition mass matrix, and the subscript indicates the sequential ordering of extraction. All columns of the extraction matrix, $A^{(1)}$, must sum to the same value. We emphasize use of the term transition mass matrix because its columns do not sum to one.
Because we cannot extract more than the minimum value at any entry, all columns of $A^{(1)}_1$ must be scaled so that their sums equal the minimum of such sums. The column sums for lag 1 are $\Sigma^M_1 = (.3, .5)$, with a minimum of $m_1 = .3$, and the column sums for lag 2 are $\Sigma^M_2 = (.6, 0)$, with a minimum of $m_2 = 0$. Thus, we can extract a transition mass matrix with total mass of .3 associated with lag 1, or a transition mass matrix with total mass of 0 associated with lag 2. We select the lag with the highest extraction mass, allowing lag 1 to provide the primary single-lag contribution to the decomposition. The extracted and leftover mass matrices are, respectively,

$$A^{(1)}_1 = \begin{pmatrix} .3 & 0 \\ 0 & .3 \end{pmatrix}, \quad B^{(1)}_1 = \begin{pmatrix} 0 & .2 & .4 & 0 \\ .4 & .2 & 0 & .4 \end{pmatrix}. \quad (9)$$

We record the lag configuration associated with the selected extraction in the length-1 vector $C^{(1)}_1 = (1)$. Applying this process again to $B^{(1)}_1$ yields $m_1 = m_2 = 0$, $A^{(1)}_2$ as a matrix of zeros, and $B^{(1)}_2 = B^{(1)}_1$, indicating no further extraction is possible using single-lag transitions. At this point, incrementing the number of lags in the extraction brings us to the order of the original tensor, and so the current $B$ becomes the final extraction, $A^{(2)}_1$. The total transition masses (normalizing sums) from each extraction provide the mixture weights. Thus, the final decomposition for this example can be represented as

$$\begin{pmatrix} 0.3 \left( \begin{array}{c} 2/3 \\ 1/3 \end{array} \right) \\ \text{intercept} \end{pmatrix} + 0.3 \left( \begin{array}{c} 1 \\ 0 \end{array} \right) + 0.4 \left( \begin{array}{c} 0.5 \\ 1.5 \end{array} \right) + 0 + \begin{pmatrix} 0.2 + 0 + 0.2 + 0 + 0.2 + 0 + 0.1 + 0.2 + 0 + 0.3 + 0.4 + 0.4 + 0 + 0.1 + 0.2 + 0 + 0.1 + 0.3 + 0.4 \end{pmatrix}, \quad (10)$$

recovering the original transition probability tensor.

### 3.2 General algorithm

Consider a general transition probability tensor $\Omega$, of order $L > 1$, for a Markov chain with $K$ states. Without loss of generality, assume the tensor is organized as in Section 2.2, with
the first mode indexing the transition distribution and always adding to one, the second
mode corresponding to values of the first lag, and so forth. The decomposition algorithm
takes as inputs $\Omega$ and pre-defined threshold $\tau_0 \in [0,1)$, and proceeds as follows:

1. Extract the intercept.
   
   (a) Compute a length-$K$ vector, denoted $m$, whose $k$th entry contains the minimum
   transition probability to state $k$ across all $K^L$ transition distributions along
   modes 2 through $L$ of $\Omega$.
   
   (b) Subtract $m$, element-wise, from all $K^L$ transition distributions along modes 2
   through $L$ of $\Omega$. Let $\mathcal{B}$ be the resulting $K^{L+1}$ transition mass “leftover” tensor.
   
   (c) Normalize the intercept by taking $\tilde{\lambda}(0) = \sum_{k=1}^{K} (m)_{k}$ and setting $\tilde{Q}(0) = (1/\tilde{\lambda}(0))m$.

2. Initialize running quantities.

   (a) Set the remaining transition total mass $\tau \leftarrow 1 - \tilde{\lambda}(0)$, which is also equal to all
   sums across the first mode of $\mathcal{B}$.
   
   (b) Set the current level (i.e., number of active lags), $r \leftarrow 1$.
   
   (c) Set the extraction sequence index, $j \leftarrow 1$.

3. While $r < L$ and $\tau > \tau_0$, repeat the following:

   (a) For the current value of $r$, enumerate all possible lag configurations $(\ell_1, \ldots, \ell_r)$,
   with $1 \leq \ell_1 < \ldots < \ell_r \leq L$. Index the configurations $c \in \{1, \ldots, (\binom{L}{r})\}$. Our
   ordering convention increments the last index fastest (e.g., if $L = 5$ and $r = 3$,
   configuration $(1, 2, 4)$ is immediately followed by $(1, 2, 5)$, then $(1, 3, 4)$).

   (b) For $c = 1, \ldots, (\binom{L}{r})$, do the following:
      
      i. Suppose lag configuration $c$ is $(\ell_1, \ldots, \ell_r)$. Calculate $\mathcal{M}_c$, a $K^{r+1}$ array of
         minima across all modes not in $\{1, \ell_1 + 1, \ldots, \ell_r + 1\}$ from the current $\mathcal{B}$.
      
      ii. Calculate a $K^r$ array, denoted as $\Sigma_c^M$, by taking sums along the first mode
          of $\mathcal{M}_c$. 

16
iii. Calculate the minimum of all values in $\Sigma_c^M$, denoted as $m_c$.

(c) Find $c^* \leftarrow \min(\{c : m_c \geq m_{c'} \forall c' \in \{1, \ldots, \binom{L}{r}\}\})$, the index corresponding to the largest total mass $m_c$. The outer minimum selects the first such index in the event of a tie.

(d) If $m_{c^*} > 0$, extract the mixture component corresponding to the selected lag configuration.

i. Set $\tilde{\lambda}^{(r)}_j = m_{c^*}$.

ii. Set the transition tensor $\tilde{Q}^{(r)}_j$ equal to the normalization of $M_{c^*}$ along its first mode, by dividing all entries by corresponding entries in $\Sigma_{c^*}^M$.

iii. Update the leftover transition mass tensor. Set $B \leftarrow B - A$, where the extraction tensor $A$ is obtained by “broadcasting” $\tilde{\lambda}^{(r)}_j \tilde{Q}^{(r)}_j$ over omitted modes to its corresponding full, order-$L$ transition probability tensor.

iv. Update $\tau \leftarrow \tau - \tilde{\lambda}^{(r)}_j$.

v. Set $C^{(r)}_j = (\ell_1^*, \ldots, \ell_r^*)$ to record the selected lag configuration $c^*$.

vi. Increment $j \leftarrow j + 1$.

(e) If $m_{c^*} = 0$, then

i. Increment $r \leftarrow r + 1$.

ii. Reset $j \leftarrow 1$.

4. Assign the final mixture component.

(a) Set $\tilde{\lambda}^{(L)} = \tau$ (which is also equal to all sums across the first mode of $B$).

(b) Set $\tilde{Q}^{(L)} = (1/\tilde{\lambda}^{(L)})B$.

Let $R$ collect levels $r \in \{1, \ldots, L - 1\}$ utilized by selected components of the algorithm, and $\{J_r : r \in R\}$ count the number of components selected at each level. For a given value of $\tau_0$, the output of the algorithm is fully characterized by the quantities $\tilde{\lambda}^{(0)}$, $\tilde{Q}^{(0)}$, $\{\tilde{\lambda}^{(r)}_j\}$, $\{\tilde{Q}^{(r)}_j\}$, $\tilde{\lambda}^{(L)}$, $\tilde{Q}^{(L)}$, and $\{C^{(r)}_j\}$, for $j = 1, \ldots, J_r$, $r \in R$. The decomposition of the full
transition tensor into its mixture form is given by

\[
\Pr(S_t = k_0 | S_{t-1} = k_1, \ldots, S_{t-L} = k_L) = (\Omega)_{k_0,k_1,\ldots,k_L} = \tilde{\lambda}(0)(\tilde{Q}(0))_{k_0} + \sum_{r \in R} \sum_{j=1}^{J_r} \tilde{\lambda}_j^{(r)}(\tilde{Q}_j^{(r)})_{k_0,k_{C(r,j,1)},\ldots,k_{C(r,j,r)}} + \tilde{\lambda}(L)(\tilde{Q}(L))_{k_0,k_1,\ldots,k_L},
\]

where the \(C(r,j,i)\) function returns the \(i\)th entry of \(C_j^{(r)}\). We draw attention to a few key differences between the decomposition in (11) and the MMTD model specification in (2). First, note that the level of interaction (number of active lags) is not restricted by an upper bound \(R < L\). Second, each \(\tilde{Q}_j^{(r)}\) within level \(r\) can be distinct, whereas the MMTD model allows only one transition probability tensor per level. Both differences reflect that the decomposition is a post-estimation tool and not a model with parameters to be estimated. Finally, it is trivial to introduce level-specific weights \(\tilde{\Lambda}_r = \sum_{j=1}^{J_r} \tilde{\lambda}_j^{(r)}\) and re-normalize the within-level weights to \(\tilde{\lambda}_j^{(r)}/\tilde{\Lambda}_r\).

The threshold input \(\tau_0\) allows a user to truncate the approximation of \(\Omega\) to a desired accuracy, such that the algorithm will cease adding levels if approximated transition probabilities differ from the originals by no more than \(1 - \tau_0\). In this case, \(\tilde{\lambda}(L)\) and \(\tilde{Q}(L)\) provide an approximation residual mass and transition tensor, which can possibly be further decomposed. The residual weight \(\tilde{\Lambda}_L\) can, and often does, exceed \(\tau_0\).

Step 3 of the algorithm can be made more efficient if lag configurations for previously extracted components are not reconsidered, as their \(m_c\) value thereafter equals zero. We omit this detail from the algorithm for simplicity.

The decomposition resulting from the proposed algorithm is unique up to the decision rule governing the sequence of extraction in Step 3c. The approach of selecting lag configurations by maximizing total extraction mass is appealing for interpretability and consistency with the MMTD model specification in Section 2.2. However, if multiple \(m_c\) values are close to the maximum, lag configurations nearly as influential as the one selected for extraction may not feature prominently in the decomposition, resulting in sensitive output that potentially masks alternate dependence structures. Applying the decomposition to estimated transition tensors across MCMC samples helps alleviate this issue. Another
robust alternative, that precludes uniqueness of the algorithm, is to make Step 3c stochastic (e.g., drawing $c^*$ from a discrete distribution with weights proportional to $m_c$).

Another consequence of the chosen decision rule in Step 3c is that decompositions of MMTD-specified tensors mixing over multidimensional lag configurations can include levels higher than $R$. Small weights typically accompany these additional levels, which have little practical importance.

### 3.3 Uses of the decomposition

The proposed decomposition builds any transition probability tensor as an accumulating sum of sequential contributions, along slices of the tensor, from component transition mass tensors of increasing order. Equivalently, this sum can be represented as a finite mixture of transition probability tensors of increasing order. Each mixture weight is interpretable as the maximum possible change to any transition probability affected by the lag configuration corresponding to the mixture component. A total contribution of each lag can be calculated by aggregating all weights of mixture components involving the lag in question.

The most informative and useful outputs from the algorithm are the set of weights, $\tilde{\lambda}$, and corresponding lag configurations, $C$. These can be used to understand how well lower-order dynamics approximate the full transition tensor, or as a diagnostic in MMTD modeling. For example, one can check model or lag-inclusion weights against those obtained by applying the algorithm to model estimates.

Absent a set of constraints characterizing an MMTD parameterization as fully “reduced”, we recommend applying the decomposition algorithm to help with identifiability of estimates from all fitted MMTD models. Inferences regarding lag relevance and mixture-component transition probabilities are likely more reliable when fully reduced.

Finally, we note that the decomposition can be applied to any transition probability tensor, extending its utility beyond the current modeling scenario. Specifically, it could be used to interpret results from any method wherein a model provides at least a point estimate of a full transition probability tensor.
4 Simulation study

To demonstrate the effectiveness of the MMTD model for both objectives and to compare transition probability estimation performance with existing methods, we report two simulation studies. Both simulation scenarios feature time series generated from Markov chains of differing order and lag configuration. In Simulation 1, the generating model has three states ($K = 3$) for which transition probabilities depend on lags 1, 3, and 4; thus the effective order is three. In Simulation 2, the generating model is a fifth-order binary chain ($K = 2$) for which all lags contribute to transition probabilities. In both models, each distribution in the transition tensor $\Omega$ was drawn from a uniform distribution on the simplex (i.e., symmetric Dirichlet distributions with all shape parameters equal to 1). Each chain was randomly initialized and run for 1,000 transitions. The first 1,000 samples thereafter were reserved for training data and the next 1,000 for validation.

To evaluate estimation of transition probabilities, we fit each model using a subset of training samples and compared point estimates of the transition distributions to the true transition distributions at each validation time. Specifically, for validation time point $t'$, each model produced a vector $\hat{p}_{t'}$ to estimate each $p_{t'}^{(k)} = \text{Pr}(s_{t'} = k \mid s_{t'-1}, \ldots, s_{t'-L}) = (\Omega)_{k,s_{t'-1},\ldots,s_{t'-L}}$, for $k = 1, \ldots, K$. In Bayesian models, the point estimate is the Monte Carlo-computed posterior mean of $\hat{p}_{t'}$. In non-Bayesian models, $\hat{p}_{t'}$ is computed from the optimized model fit. For each validation time point, we computed the $L_1$ loss given by $L_{t'} = \sum_{k=1}^{K} |\hat{p}_{t'}^{(k)} - p_{t'}^{(k)}|$. The reported loss metric for model comparison is $100 \times \sum_{t'} L_{t'}/(KT')$, i.e., 100 times the mean $L_1$ loss across the $T' = 1,000$ validation points.

We fit MMTD models to each training set with different settings of the maximum interaction level $R$. Let MMTD($L, R$) denote a model fit with user-specified maximum lag horizon (full model order) $L$, and maximum interaction level $R$. Although the priors have different settings to accommodate various beliefs, we recommend the following default choices used for these simulation studies. All transition distributions in all $\{Q\}$ utilize symmetric, unit-information Dirichlet priors (i.e., whose shape parameters all equal $1/K$ so that they sum to one). The priors for $\{\lambda^{(r)}\}$ use Dirichlet shape parameters $\alpha_{\lambda^{(r)}} = (1/(T), \ldots, 1/(T))$, and SDM sparsity $\beta_{\lambda^{(r)}} = \sqrt{T}$ to encourage selection of a single-lag
configuration within level \( r \). The SBM prior for \( \Lambda \) uses \( \pi_1 = 0 \) for \( \Lambda_0 \) and \( \pi_1 = 0.25 \) thereafter; \( \pi_3 = 0 \) for \( \Lambda_0 \) and \( \pi_3 = 0.25 \) thereafter; \( \eta = 1,000 \); and \( \gamma = \delta = 1 \) for all second-component beta distributions, yielding a uniform prior for \( \Lambda_0 \). This prior avoids penalizing \( \Lambda_0 \), allows for sparsity in the remaining lags, and maintains soft ordering that favors lower levels of the model. Because \( R \) is typically kept to small values, it is important that \( \pi_1 \) not be large and that \( \pi_3 \) not be too small. Otherwise, the prior can inappropriately allocate substantial mass toward large values of \( \Lambda_R \). We recommend checking for this condition as part of prior sensitivity analysis.

To obtain results in Sections 4.1, 4.2, and 5, each model was initialized with random draws from Dirichlet distributions for \( \Lambda \) and each \( \lambda^{(r)} \). Random initialization in these models calls for long burn-in periods, on the order of tens to hundreds of thousands of iterations. In our analyses, 200,000 burn-in iterations were followed by another 400,000 iterations. Reported posterior quantities were calculated using a thinned sample retaining every 200th iteration from one of the replicate chains. In a few cases, parallel chains explored different modes, impacting inferences and performance. Most post-processing and graphics for MMTD model fits were generated using the R language (R Core Team, 2019) with the \textit{coda} (Plummer et al., 2006) and \textit{ggplot2} (Wickham, 2016) packages.

In addition to the proposed model, we fit multinomial generalized linear models with logistic link functions using the \textit{VGAM} package in R (Yee, 2010). To distinguish different settings, we denote the model fit as LogitMC\((L, R')\), with maximum lag horizon \( L \) and highest interaction order among the linear predictors \( R' \). To allow for higher levels of interaction, we also fit lasso-penalized multinomial logit models with leave-one-out cross validation, denoted with LogitMC\((L, R', \text{lso})\), using the \textit{glmnet} package in R (Friedman et al., 2010). We also fit variable length Markov chain models, denoted VLMC, using the \textit{VLMC} package in R (Maechler, 2019) and employing default model settings. Note that the MMTD\((L, 1)\) model is equivalent to an MTD model with an intercept term and SDM prior on the mixture weights.
Table 4: Simulation 1 (three-state chain with active lags 1, 3, and 4). (Left) Mixture decomposition profile for the simulation-generating tensor. (Right) Results for transition probability estimation under various models/settings using sample sizes $T = 200$ and $T = 500$. The lowest mean loss is highlighted with bold font.

<table>
<thead>
<tr>
<th>Level</th>
<th>Weight ($\tilde{\lambda}$)</th>
<th>Lag config. ($C$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.040</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>.086</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>.015</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>.116</td>
<td>1, 3</td>
</tr>
<tr>
<td>2</td>
<td>.033</td>
<td>1, 4</td>
</tr>
<tr>
<td>3</td>
<td>.710</td>
<td>1, 3, 4</td>
</tr>
<tr>
<td>4</td>
<td>.000</td>
<td>1, 2, 3, 4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lag</th>
<th>Weight (total contribution)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.944</td>
</tr>
<tr>
<td>2</td>
<td>.000</td>
</tr>
<tr>
<td>3</td>
<td>.841</td>
</tr>
<tr>
<td>4</td>
<td>.743</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>100 $\times$ Mean $L_1$ loss</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$T = 200$</td>
</tr>
<tr>
<td>VLMC</td>
<td>19.12</td>
</tr>
<tr>
<td>LogitMC(6, 1)</td>
<td>18.70</td>
</tr>
<tr>
<td>LogitMC(6, 2)</td>
<td>37.42</td>
</tr>
<tr>
<td>LogitMC(6, 3)</td>
<td>n/a</td>
</tr>
<tr>
<td>LogitMC(6, 1, lso)</td>
<td>17.70</td>
</tr>
<tr>
<td>LogitMC(6, 2, lso)</td>
<td>16.52</td>
</tr>
<tr>
<td>LogitMC(6, 3, lso)</td>
<td>16.82</td>
</tr>
<tr>
<td>LogitMC(6, 5, lso)</td>
<td>16.85</td>
</tr>
<tr>
<td>MMTD(6, 1)</td>
<td>17.29</td>
</tr>
<tr>
<td>MMTD(6, 2)</td>
<td>14.76</td>
</tr>
<tr>
<td>MMTD(6, 3)</td>
<td>$$\textbf{13.86}$$</td>
</tr>
<tr>
<td>MMTD(6, 5)</td>
<td>14.36</td>
</tr>
</tbody>
</table>

4.1 Simulation 1 results

Because the generating transition probability tensor is available in simulation studies, we use the decomposition of $\Omega$ directly as a benchmark for MMTD models. The decomposition profile for Simulation 1 is summarized in Table 4 (left). The mixture component corresponding to the true lag configuration (1, 3, 4) dominates the weights. However, lower-level components depending on lag 1 only and lags 1 and 3 also carry appreciable weight. In this particular simulation, lag 4 has the weakest signal of the active lags, and lag 2 has no associated mass, as expected. The full generating model has 54 free parameters.

All models were fit to the first $T = 200$, and $T = 500$ training samples of the Simulation 1 time series. Here, we assume that the modeler is considering a horizon of six lags ($L = 6$), which we use where possible to promote equitable comparisons. Results of the mean $L_1$ loss across the 1,000 validation points are given in Table 4 (right). In addition to transition probability estimation, we are interested in inferences for lag relevance afforded by the MMTD models.
4.1.1 Sample size 200

In the $T = 200$ case, the variable length Markov chain model performs poorly. The multinomial logistic class is unable to handle anything beyond the single-lag additive model with the short series. Fitting all second-level interactions for up to six lags is cumbersome, and estimation fails for third-level interactions. The lasso penalty stabilizes the logit model, allowing higher-level interactions without degrading performance.

Several MMTD models were fit with increasing interaction level $R$ ranging from 1 to 5. The model with $R = 2$ provides a substantial improvement over mixing single-lag transitions and fits nearly as well as models that include additional levels. As anticipated, estimation performance stops improving when $R$ exceeds the true effective order of three.

In all MMTD models with $R \geq 2$, posterior inferences most strongly support level two, specifically lag configuration (3, 4). Models allowing a third level most often select the correct configuration (1, 3, 4). The SBM prior on $\Lambda$ shrinks inferences toward level two, but not decidedly away from the third. Overall, the MMTD consistently produces the most faithful estimates of transition probabilities from a single model.

It is evident that with this shorter time series, there is little to be gained by utilizing interactions beyond two lags, regardless of the modeling approach. We note that models incorporating shrinkage (lasso-LogitMC and MMTD) do not suffer significant penalties for over-specification (i.e., $R' > 3$ or $R > 3$ in this case).

4.1.2 Sample size 500

With a larger sample size, the VLMC model is more competitive. The multinomial logit model better handles more parameters, but still fails to extend to three-lag interactions. Regularization again admits higher-level interactions with consistent performance.

In this large-sample scenario, MMTD performance improves substantially when the specified mixture model contains the generating model structure (i.e., $R \geq 3$), strengthening the case for the first proposed use of the MMTD model. When $R \geq 3$, posterior mass concentrates on the correct lag configuration, with no drop in performance when the interaction level is over-specified. Lower levels primarily mix over lag configurations (3)
Table 5: Simulation 2 (fifth-order, binary chain). (Left) Mixture decomposition profile for the simulation-generating tensor. (Right) Results for transition probability estimation under various models/settings using sample sizes $T = 100$, $T = 200$, and $T = 500$. The lowest mean loss is highlighted with bold font.

<table>
<thead>
<tr>
<th>Level</th>
<th>Wt. ($\tilde{\lambda}$)</th>
<th>Lag config. ($\mathcal{C}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.066</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>.026</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>.043</td>
<td>2, 4</td>
</tr>
<tr>
<td>2</td>
<td>.040</td>
<td>1, 2</td>
</tr>
<tr>
<td>3</td>
<td>.027</td>
<td>1, 2, 4</td>
</tr>
<tr>
<td>3</td>
<td>.009</td>
<td>3, 4, 5</td>
</tr>
<tr>
<td>4</td>
<td>.155</td>
<td>2, 3, 4, 5</td>
</tr>
<tr>
<td>4</td>
<td>.014</td>
<td>1, 2, 4, 5</td>
</tr>
<tr>
<td>4</td>
<td>.003</td>
<td>1, 3, 4, 5</td>
</tr>
<tr>
<td>4</td>
<td>.002</td>
<td>1, 2, 3, 4</td>
</tr>
<tr>
<td>5</td>
<td>.616</td>
<td>1, 2, 3, 4, 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lag</th>
<th>Weight (total contrib.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.702</td>
</tr>
<tr>
<td>2</td>
<td>.897</td>
</tr>
<tr>
<td>3</td>
<td>.784</td>
</tr>
<tr>
<td>4</td>
<td>.868</td>
</tr>
<tr>
<td>5</td>
<td>.823</td>
</tr>
</tbody>
</table>

and (3, 4). Again, among the models considered in this simulation scenario, the MMTD consistently produces the most faithful estimates of transition probabilities.

4.2 Simulation 2 results

The mixture decomposition profile for Simulation 2 given in Table 5 (left) reveals full, fifth-order dynamics in which all lags interact. At most one subset of lags carries appreciable weight. The full generating model has 32 free parameters.

All models were fit to the first $T = 100$, $T = 200$, and $T = 500$ training samples from Simulation 2. Here, we assume that the modeler is considering a horizon of seven lags, which we use where possible to promote equitable comparisons. Results of the mean $\mathcal{L}_1$ loss across the 1,000 validation points are given in Table 5 (right).

4.2.1 Sample size 100
The VLMC model performs well for all sample sizes in this scenario, and best among all methods for $T = 100$. This is mildly surprising, given that the generating tensor is not sparse, exhibiting full, fifth-order dependence. In the $T = 100$ case, higher-level interactions are not estimable in the multinomial logit model. Models with lower-level interactions perform comparably across model classes. Again, regularization preserves integrity of over-specified models.

MMTD models tend to mix with lag 2 (and occasionally lag 4) in the first level. They tend not to support higher levels, and fail to coalesce around a best two, three, or four-lag configuration. It appears that the prior structure holds against a weak signal, as the sample size is not sufficiently large to estimate fifth-order dynamics.

### 4.2.2 Sample size 200

The larger sample size helps the models reveal higher-order dynamics. The VLMC model is again competitive, and generally outperforms multinomial logit models. Interactions significantly improve logit models, which marginally outperform their MMTD counterparts in this scenario. Low-level MMTD models are unable to leverage increased sample size to the extent that the other models can.

The MMTD(7, 2) model leverages its mixture structure to include lags 1 and 4 on the first level and lag configuration (3, 5) on the second level, yielding an effective order between 3 and 4. The MMTD(7, 4) model mixes primarily over lag configurations (1, 2, 3, 5), (1, 2, 3, 6), (1, 2, 3), and (3, 5). The over-specified MMTD(7, 7) model clearly identifies the fifth level and correct lag configuration, and mixes with a fourth-level configuration (1, 2, 3, 5).

### 4.2.3 Sample size 500

The 32 free parameters are more easily identified from the series with 500 samples. Here, the VLMC model is less competitive, but still performs well. Again, interaction significantly improves multinomial logit model fits, while low-level mixing in MMTD models fails to approximate the dependence structure.
As in the $T = 200$ case, the MMTD(7, 2) model mixes over non-overlapping lag configurations, leveraging the MTD structure in a manner consistent with the model’s second intended use. However, it does not yield strong performance gains in this scenario.

The over-specified MMTD(7, 7) model again clearly identifies the true structure, and in this case performs best. Replicate runs of the MMTD(7, 4) model yield losses well below the reported 15.28 (12.12 and 9.73), indicating that MCMC chains are exploring distinct posterior modes. For this reason, we recommend always running multiple parallel chains.

5 Data illustrations

We now apply the MMTD model to two data analyses. The first example was studied with the original MTD and in the subsequent literature. The second is an analysis of pink salmon population dynamics in Alaska, U.S.A. in the mid-20th century. We focus on illustrating inferences for lag relevance available from the MMTD model.

5.1 Seizure data

Berchtold and Raftery (2002) demonstrate the MTD model using a binarized time series adapted from MacDonald and Zucchini (1997), which reports the occurrence of at least one epileptic seizure for a patient on each of 204 consecutive days. Berchtold and Raftery (2002) fit several Markov chain and MTD models, using the Bayesian information criterion (BIC) to ultimately select an MTD with eight lags. They report that $\lambda_8$ has the greatest magnitude. Note that the MTD model used in Berchtold and Raftery (2002) allows negative values in $\lambda$, which requires a complex set of constraints for estimation. The seizure time series was revisited and fit using the methods in Sarkar and Dunson (2016), who report a model of maximal order 8, with lag 8 having the highest posterior inclusion probability. In contrast with Berchtold and Raftery (2002), they find lag 1 to be the second most important. They report the posterior mode for the number of important lags to be three.

In light of these two analyses, we fit the MMTD model to the seizure data with $L = 10$ and $R = 4$, using identical prior settings to those in the simulation studies. Trace plots
(not shown) indicate that the marginal posterior distributions over $\Lambda$ and each $\lambda^{(r)}$ are multimodal, suggesting that more than one combination of lags could model the dynamics with similar accuracy. In such cases, it is important to run multiple long MCMC chains. We note also that the assumption of time-homogeneity is questionable, as no seizures were reported in the last 29 days.

Posterior inferences for level weights, supported by marginal densities in Figure 1 and pairwise scatter plots of samples for $\Lambda$ (not shown), reveal no clear selection of any interaction level. Densities for aggregated level weights, computed by applying the mixture decomposition to the estimated transition probability tensor for each posterior sample, are included as dashed lines. Note that the decomposed level weights exhibit less bimodality in this analysis, possibly a consequence of added stability from the decomposition being more versatile and better identified than the model parameters. Within-level configuration weights in $\{\lambda^{(r)}\}$ support lags 4 and 8 at the first level and lag configuration (8, 9) at the second level, but no particular three or four-lag configurations stand out (with exception

![Figure 1: Marginal posterior density plots for level weights in the seizure analysis from the fitted MMTD(10, 4) model (solid lines) and from the mixture decomposition of the same fit (dashed lines).](image)
of configuration (7, 8, 9) in one MCMC chain).

Without clear lag selection, and because of discrepancies between model estimates and decomposition estimates, we discourage over-interpretation of component tensors \{Q^{(r)}\}.

However, in this case they do reveal that transition probabilities favor persisting in states. For example, the posterior means for \((Q^{(2)})_{1,1,1}\) and \((Q^{(2)})_{2,2,2}\) are 0.76 and 0.72, respectively (posterior medians are 0.90 and 0.85). That is, absence of seizures in recent days yields a high probability for no seizure on the current day, and repeated occurrence of seizures on multiple past days yields a high probability of seizure on the current day.

We can more comprehensively assess lag relevance by computing an aggregated inclusion weight as the sum of all products \(\Lambda_r \times \lambda^{(r)}_{(z_j)}\) across \(j = 1, \ldots, (L_r)\) and \(r = 1, \ldots, R\) for which lag \(\ell\) appears in the lag configuration \(z_j\). An analogous calculation for the mixture decomposition was noted in Section 3.3. We compute this aggregate weight for each lag at each MCMC sample. Inference for \(\Lambda_0\) is included as lag 0 for reference. A high inclusion weight for lag 0 should not be interpreted as a lack of Markovian dependence (unless it is near 1 with high confidence). However, a low inclusion weight for lag 0 relative to other lags can indicate strong dependence.

We summarize aggregated inclusion weights for the model fit and decomposition in Figure 2. The decomposition-based weights noticeably shift additional weight to the intercept, but inferences for lag relevance are qualitatively unchanged. Despite large uncertainty, lags 4, 8, and 9 stand out. We note that the posterior inclusion pattern across lags resembles

![Figure 2: Marginal posterior densities of aggregated lag-inclusion weights in the seizure analysis from the fitted MMTD(10, 4) model (left), and from the mixture decomposition of the same fit (right). Posterior medians appear as points, and 95% intervals as vertical lines.](image-url)
a plot with similar interpretation in Figure 6(e) of Sarkar and Dunson (2016). The most notable exceptions are that lags 1 and 9 trade prominence between the models.

5.2 Pink salmon data

We next investigate a time series of annual pink salmon abundance (escapement) at a creek in Alaska, U.S.A. from 1934 to 1963 (Alaska Fisheries Science Center, 2018). Population dynamics for pink salmon provide a testing opportunity for our model because pink salmon have a strict two-year life cycle (Heard, 1991). Thus, we expect even lags to have the most influence in predicting the current year’s population. A time-series plot of abundance is given in Figure 3 together with bivariate lag scatter plots (the latter for the natural logarithm of abundance). In this scenario, we might expect non-stationarity with long-term trends. It appears from the time series that the even-year population began to struggle in the late 1940s. Repeated interventions throughout the 1950s culminated in a population transfer in 1964 that bisects the complete time series and restricts us to the first segment.

Figure 3: Time-series plot of pink salmon abundance from 1934 to 1963 (top) and bivariate lag plots for the natural logarithm of abundance (bottom). In the lag plots, $y_t$ denotes abundance at time $t$ and horizontal/vertical lines separate $K = 4$ quantile-based bins used to assign $\{y_t\}$ into discrete states $\{s_t\}$. 

29
Nevertheless, the lag scatter plots suggest that we should be able to detect lag dependence structure, even with as few as 30 observations. After discretizing the data into sets of $K = 4$ quantile-based bins using all 30 years, we fit the proposed model with the same prior settings used for the simulation studies. Because discretization is based on quantiles, results are invariant to monotonic transformations such as the natural logarithm.

The MMTD(4, 2) model fit splits posterior weight between levels 1 and 2 almost evenly. Uncertainty, stemming from noisy dynamics and a small sample size, again results in a multimodal posterior. This uncertainty is also apparent in posterior inferences for the lag-inclusion weights, summarized in Figure 4. In these plots, we see essential agreement between the lag-inclusion weights from the model and decomposition, with lag 2 being most prominent.

Because just one mixture component dominates selection in the first level, we can interpret component-specific transition probabilities in $Q^{(1)}$ as primarily a transition matrix from the active lag, in this case lag 2. The posterior median (for each matrix entry) is reported in the left panel of Figure 5. Lag configuration (1, 2) carries most weight in the second level, followed distantly by configuration (2, 4). Thus, the posterior median estimate of $Q^{(2)}$ reported on the right panel of Figure 5 applies primarily to transitions from those pairs of lags. Inferences for aggregated lag weights and patterns in second-level transition distributions suggest there may be slight additional dependence on lags other than just the

![Figure 4](image-url)

Figure 4: Posterior analysis of aggregated lag-inclusion weights for the MMTD(4, 2) fit of the pink salmon time series. Marginal posterior densities of aggregated inclusion weights from the model (left), and from the mixture decomposition of the same fit (right), report posterior medians as points, and 95% intervals as vertical lines.
Figure 5: Posterior median point estimates of $Q^{(1)}$ (left) and a matricized representation of $Q^{(2)}$ (right) from the MMTD(4, 2) pink salmon analysis. Rows (along the y-axis) represent states to which the transition occurs, and columns (along the x-axis) represent the states occupied by the first two selected lags, with the state corresponding to the most recent lag changing index first.

We note, however, that estimation of the second-level transition tensor does spread the short time series thin. Further note that estimated component transition tensors from the model are not identified and can differ significantly from those of the decomposition. We caution against reliance on model estimates alone.

6 Discussion

We have explored an extension of the original mixture transition distribution model for high-order Markov chains, and introduced a related mixture decomposition. Our Bayesian approach with carefully selected, structured priors admits characterization of uncertainty and facilitates two primary uses of the model: 1) model over-specification and shrinkage to identify the active dependence structure, and 2) mixing low interaction levels to approximate higher-order dynamics. The model is most performant with the first of these objectives, as we have demonstrated through comparison with other methods for transition probability estimation, as well as data analysis.

The mixture decomposition yields an interpretable representation of any transition probability tensor in terms of MTD-type components. The algorithm sequentially extracts maximal transition mass, beginning with lower levels. Because it strictly enforces conditions that the priors in the MMTD model only encourage, the decomposition can be
applied to MMTD-estimated transition tensors to aid in model checking and provide more identifiable inferences. As a post-estimation tool, the decomposition nevertheless reflects, and is limited by, the quality of the original fit.

If decomposition from a fitted MMTD model reveals several mixture components with significant weight within a single level, it may be desirable to specify an MMTD-type model with these characteristics, i.e., by reducing $R$ and allowing distinct component transition tensors within each level. Berchtold and Raftery (2002) refer to the special case where $R = 1$ as the MTDg model. While the proposed decomposition produces members of what could be termed the MMTDg class, parameterizing a model in this way is impractical for all but very low values of $K$, $L$, and $R$. Nevertheless, it would be interesting to more fully characterize identifiable members of the MMTDg class, and incorporate this information directly into the estimation procedure, similar to the approach taken by Tank et al. (2017).

The MMTD model can approximate high-order dynamics via constructive additivity among lower-order component transition probability tensors. When lower levels are inadequate, a full jump to the next interaction level in the MMTD is required, including estimation of a mixture component tensor with $K$ times the number of parameters in the previous level. A more parsimonious compromise might rely on factorization of the higher-level tensors. We do not pursue this here, but rather choose to emphasize the structure of the proposed MMTD model, which showcases an interpretable, model-averaging flavor.

**Acknowledgements**

This research was supported in part by the National Science Foundation under award SES 1631963. The authors gratefully acknowledge contributions from Stephan Munch. They also wish to thank two reviewers, an associate editor, and the editor for several helpful comments. This research is part of the Ph.D. dissertation of M. Heiner, completed at University of California, Santa Cruz.
A MCMC algorithm details

Following the hierarchical MMTD model outlined in (5), the joint posterior distribution of all unknown parameters is given up to proportionality:

\[ p(\{\zeta_t\}_{t=L+1}^T, \Lambda, \{\lambda^{(r)}\}_{r=1}^R, \{Q^{(r)}\}_{r=0}^R, \{s_t\}_{t=L+1}^T) \propto p(\Lambda) p(Q^{(0)}) \prod_{r=1}^R \left( p(\lambda^{(r)}) \prod_{j=1}^{K_r} p\left( (Q^{(r)})_{\cdot,j} \right) \right) \times \prod_{t=L+1}^T \left[ p\left( \zeta_t \mid \Lambda, \{\lambda^{(r)}\}_{r=1}^R \right) p\left( s_t \mid \zeta_t, \{Q^{(r)}\}_{r=0}^R, \{s_{t-L}^L\}_{t=1}^L \right) \right], \]  

where \((Q^{(r)})_{\cdot,j}\) denotes column \(j\) from a matricized version of \(Q^{(r)}\).

We use a collapsed Gibbs sampler that cycles through conditional updates of \(\{\zeta_t\}, \Lambda\) and each \(\lambda^{(r)}\), with all \(\{Q^{(r)}\}\) marginalized out of the posterior. It is then straightforward to draw each \(Q^{(r)}\) from the full conditional distributions given below.

In what follows, let \(Z(\zeta)\) and \(z(\zeta)\) map \(\zeta\) to its corresponding \(Z\) and \(z\) respectively. Also, let \(\varrho_r(s)\) be a unique map from each possible length-\(r\) vector of lagged states \(s \in \{1, \ldots, K\}^r\) to the corresponding column index of the flattened (matricized) \(Q^{(r)}\). Further, let \(s_{t-1}(z)\) be a function accepting a lag configuration \(z\) and returning the values of the states at those selected lags from the vector \((s_{t-1}, s_{t-2}, \ldots, s_{t-L})\). For example, if \(z_t = (2, 5)\), then \(s_{t-1}(z_t)\) will return the vector \((s_{t-2}, s_{t-5})\).

For each \(r = 1, \ldots, R\), let \(N^{(r)}\) be a matrix containing transition counts for which the \((k, j)\) entry is the cardinality of \(\{t : Z(\zeta_t) = r \text{ and } \varrho_r(s_{t-1}(z(\zeta_t))) = j \text{ and } s_t = k\}\). Also let the \(k\)th entry of vector \(N^{(0)}\) be the cardinality of \(\{t : Z(\zeta_t) = 0 \text{ and } s_t = k\}\). Integrating all \(\{Q^{(r)}\}\) from the full joint posterior proportional to (12) yields

\[ p\left( \{\zeta_t\}, \Lambda, \{\lambda^{(r)}\} \mid \{s_t\} \right) \propto \text{SBM}(\Lambda) \prod_r \left[ \text{SDM}(\lambda^{(r)}) \right] \prod_t \left[ \Lambda_{Z(\zeta_t)} \lambda^{(Z(\zeta_t))}_{\varrho_r(\zeta_t)} \right] \times \prod_{r=1}^R \left[ p\left( \mathbf{N}^{(0)} \mid \{\zeta_t\}, \{s_t\} \right) \prod_{j=1}^{K_r} p\left( ((Q^{(r)})_{\cdot,j}) \mid \{\zeta_t\}, \{s_t\} \right) \right], \]  

where \((Q^{(r)})_{\cdot,j}\) denotes column \(j\) from a matricized version of \(Q^{(r)}\).
where
\[
p((N^{(r)}), j | \{\zeta_t\}, \{s_t\}) = \frac{\Gamma(\sum_k(\alpha_Q^{(r)})_k) \prod_k \Gamma((\alpha_Q^{(r)})_k + (N^{(r)}))_{k,j}}{\prod_k \Gamma((\alpha_Q^{(r)})_k) \Gamma(\sum_k(\alpha_Q^{(r)})_k + (N^{(r)}))_{k,j}} \; ; \quad (14)
\]
and \( p(N^{(0)} | \{\zeta_t\}, \{s_t\}) \) takes the same Dirichlet-multinomial form.

The \( \{\zeta_t\} \) are updated individually using the collapsed conditional probability mass function given by
\[
p(\zeta_t | \cdots, -\{Q^{(r)}\}) \propto \Lambda_{Z(\zeta_t)} \lambda_{Z(\zeta_t)}^{(Z(\zeta_t))} p(N^{(0)} | \{\zeta_t\}, \{s_t\}) \times \prod_{r=1}^{R} \left[ \prod_{j=1}^{K^r} p((N^{(r)}), j | \{\zeta_t\}, \{s_t\}) \right] \; ; \quad (15)
\]
where we modify \( \{N^{(r)}\} \) to reflect each possible \( \zeta_t \in \{0, 1, \ldots, \lfloor (L^r_1 + (L^r_2) + \ldots + (L^r_R)) \rfloor \} \). Instead of sampling from (15) directly, we Metropolize the update by first drawing a candidate with probability mass proportional to (15), excluding the current state. The Metropolis acceptance ratio is then the sum over all probabilities excluding the current state, divided by the sum over all full conditional probabilities excluding the candidate state (Liu, 1996; Robert and Casella, 2004, p. 394).

The remaining full conditional distributions do not change with marginalization:

- \( p(\Lambda | \cdots) \propto p(\Lambda) \prod_t p(\zeta_t | \Lambda, \{\lambda^{(r)}\}) \propto \text{SBM}(\Lambda; \pi_1, \pi_3, \eta, \gamma, \delta) \prod_t \Lambda_{Z(\zeta_t)} \), a conjugate SBM-multinomial update using the counts of \( Z(\zeta_t) \) in each of \( \{0, 1, \ldots, R\} \). See Section 2.2 of Heiner et al. (2019) for a sampling strategy based on the stick-breaking construction, noting that in the MMTD model application, \( \pi_1 \) and \( \pi_3 \) change with index \( r = 0, 1, \ldots, R \).

- \( p(\lambda^{(r)} | \cdots) \propto p(\lambda^{(r)}) \prod_t p(\zeta_t | \Lambda, \{\lambda^{(r)}\}) \propto \text{SDM}(\lambda^{(r)}; \alpha^{(r)}_\lambda, \beta^{(r)}_\lambda) \prod_{t:Z(\zeta_t)=r} \lambda_{Z(\zeta_t)}^{(r)} \) independently for \( r \in \{1, \ldots, R\} \). Here, \( \lambda^{(r)} \) is indexed by the \( \binom{L_r}{r} \) possible sets of lags. This is a conjugate SDM-multinomial update using the counts of the \( \binom{L_r}{r} \) unique lag configurations \( z_t \) within level \( r \). The full conditional is a SDM distribution with \( \beta^{(r)}_\lambda \) and with the multinomial counts added to \( \alpha^{(r)}_\lambda \), analogous to Dirichlet full conditionals.
\[ p(\mathbf{Q}(0) | \ldots) \propto p(\mathbf{Q}(0)) \prod_{t:Z(\zeta_t)=0} p \left( s_t \mid \zeta_t, \{\mathbf{Q}(\zeta)^{r}_{\tau=0}, \{s_{t-\ell}\}_{\ell=1}^L \right) \]

\[ = \mathrm{Dir}(\mathbf{Q}(0) | \alpha_{Q(0)}) \prod_{t:Z(\zeta_t)=0} (\mathbf{Q}(0))_{st} \] a standard conjugate Dirichlet-multinomial update using the counts collected in \( N(0) \). The full conditional distribution is then \( \mathrm{Dir}(\alpha_{Q(r)} + N(0)). \)

\[ p \left( \left( \mathbf{Q}(r) \right)_{.,j} \mid \ldots \right) \propto p \left( \left( \mathbf{Q}(r) \right)_{.,j} \right) \prod_{t:Z(\zeta_t)=r \text{ and } \varrho_{r}(s_{t-1}(z(\zeta_t)))=j} \times \]

\[ p \left( s_t \mid \zeta_t, \{\mathbf{Q}(r)^{r}_{\tau=0}, \{s_{t-\ell}\}_{\ell=1}^L \right) \]

\[ \propto \mathrm{Dir} \left( \left( \mathbf{Q}(r) \right)_{.,j} \mid \alpha_{Q(r)} \right) \prod_{t:Z(\zeta_t)=r \text{ and } \varrho_{r}(s_{t-1}(z(\zeta_t)))=j} (\mathbf{Q}(r))_{st,j} \]

independently for \( r = 1, \ldots, R \), and \( j = 1, \ldots, K^r \). Again, this is a standard conjugate Dirichlet-multinomial update using the transition counts collected in \( (N(r))_{.,j} \). The full conditional distribution is then \( \mathrm{Dir}(\alpha_{Q(r)} + (N(r))_{.,j}). \)

SUPPLEMENTARY MATERIAL

MTD examples: Directory containing code and data necessary to run the simulation and pink salmon analyses using the MMTD models with the MTD Julia package (available at https://github.com/mheiner/MTD.jl.git). The file README.md contains descriptions and instructions. The seizure data are available in Berchtold and Raftery (2002). (The MTD_examples directory can be downloaded from https://github.com/mheiner/MTD_examples.git)

References


