Multistate Analysis with Infinite Mixtures of Markov Chains Supplementary Material

Lucas Maystre[1] **Tiffany Wu**[1] **Roberto Sanchis-Ojeda**[1] **Tony Jebara**[1]

[1]Spotify

A DISCRETE-TIME MODEL

In this appendix, we provide further details on the discrete-time model. We begin by discussing some of the properties of the resulting compound process. Then, we derive a simpler variant of the model by replacing the generalized Dirichlet mixture distribution by a (standard) Dirichlet distribution.

A.1 PROPERTIES OF THE COMPOUND PROCESS

It is insightful to contrast the properties of a Markov chain with those of our compound process. Unlike a DTMC, our model no longer satisfies the Markov property, and in general future transitions depend on the entire past. While (ergodic) DTMCs converge to a stationary distribution [Norris, 1998], our compound process does not: By construction, the distribution our process converges to is different for different values of the latent $\boldsymbol{\Theta}$. Nevertheless, it is easy to show (by continuity) that the limiting distribution

$$\lim_{k\to\infty} \int \boldsymbol{\pi}^\top \boldsymbol{\Theta}^k p(\boldsymbol{\Theta}) d\boldsymbol{\Theta}$$

exists and is independent of the initial distribution $\boldsymbol{\pi}$. We also note that Markov chains are a limiting case of our compound process. Informally, this happens when the mixture distribution concentrates at a single value of $\boldsymbol{\Theta}$. We make this precise in the next section..

A.2 SIMPLIFIED DISCRETE-TIME MODEL

Preliminaries. The Dirichlet distribution has support on the set of N-dimensional probability vectors and density

$$\mathrm{Dir}(\boldsymbol{x} \mid \boldsymbol{\eta}) = \frac{1}{B(\boldsymbol{\eta})} \prod_{i=1}^{N} x_i^{\eta_i - 1},$$

where $\boldsymbol{\eta} \in \mathbf{R}^N_{>0}$ is a parameter vector and the multivariate beta function is defined as $B(\boldsymbol{\eta}) = \prod_i \Gamma(\eta_i) / \Gamma(\sum_i \eta_i)$.

It is easy to verify that $\mathrm{Dir}(\boldsymbol{x} \mid \boldsymbol{\eta}) = \mathrm{GDir}(\boldsymbol{x} \mid \boldsymbol{\alpha}, \boldsymbol{\beta})$ if $\alpha_i = \eta_i$ and $\beta_i = \sum_{j=i+1}^{N} \eta_j$ for $i = 1, \ldots, N-1$ [Connor and Mosimann, 1969]. The Dirichlet distribution can be reparametrized by a concentration parameter $\rho = \sum_i \eta_i$ and a mean vector $\bar{\boldsymbol{\eta}}$, where $\bar{\eta}_i = \eta_i / \rho$. For any mean vector $\bar{\boldsymbol{\eta}}$, we have that

$$\lim_{\rho\to\infty} \mathrm{Dir}(\boldsymbol{x} \mid \rho\bar{\boldsymbol{\eta}}) = \delta(\boldsymbol{x} - \bar{\boldsymbol{\eta}}), \tag{1}$$

where δ is the Dirac delta function. That is, the distribution concentrates around its mean $\bar{\boldsymbol{\eta}}$ as ρ becomes larger.

Mixture Model. We assume that the transition matrix $\boldsymbol{\Theta}$ of a DTMC is sampled from a product of Dirichlet distributions,

$$p(\boldsymbol{\Theta} \mid \boldsymbol{H}) = \prod_i \mathrm{Dir}(\boldsymbol{\theta}_i \mid \boldsymbol{\eta}_i),$$

where $\boldsymbol{H} = [\boldsymbol{\eta}_i]$. In other words, each row of $\boldsymbol{\Theta}$ is sampled from a distinct Dirichlet distribution independently from the other rows. We can write the compound likelihood given a sequence s as

$$p(s \mid \boldsymbol{H}) = \int p(s \mid \boldsymbol{\Theta}) p(\boldsymbol{\Theta} \mid \boldsymbol{H}) d\boldsymbol{\Theta} = \prod_{i=1}^{N} \frac{B(\boldsymbol{\eta}_i + \boldsymbol{k}_i)}{B(\boldsymbol{\eta}_i)},$$

where $\boldsymbol{K} = [k_{ij}]$ is the matrix counting the number of transitions observed between each pair of states. Note that the Dirichlet mixture model has N^2 free parameters, compared to $N(N-1)$ for a DTMC and $2N(N-1)$ for the generalized mixture model.

DTMC as a Limiting Case. Let $\bar{\boldsymbol{\Theta}}$ be a (row-stochastic) transition matrix, and let $\boldsymbol{H} = \rho\bar{\boldsymbol{\Theta}}$ for some $\rho > 0$. Then, by property (1),

$$p(s \mid \boldsymbol{H}) \xrightarrow{\rho\to\infty} p(s \mid \bar{\boldsymbol{\Theta}}) = \prod_{i,j} \bar{\theta}_{ij}^{k_{ij}},$$

which shows that a DTMC is the limiting case of a Dirichlet mixture model when the concentration parameter tends to infinity.

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

B PREDICTIVE STATE DISTRIBUTION

In this appendix, we first develop polynomial-time algorithms for computing the exact predictive state distribution in the discrete-time setting. Then we provide the continuous-time equivalent of Algorithm 1 presented in the main text. We conclude with an empirical study of Algorithm 1.

B.1 DIRECTED ACYCLIC GRAPHS WITH SELF-LOOPS

We assume that the transition graph $\mathcal{G} = ([N], \mathcal{E})$ is such that there are no directed cycles, except for self-loops. Intuitively, this restriction means that once the process leaves a state, it will never return to that state in the future. This special case is relevant in practice: For example, the EBMT dataset we consider in Section 5 of the main text satisfies the assumption.

For conciseness, we consider only the (simple) Dirichlet mixture model introduced in Appendix A. The extension to the generalized Dirichlet distribution is straightforward. We assume that the parameter matrix $\boldsymbol{H} = [\eta_{ij}]$ is such that $\eta_{ij} = 0$ if $(i, j) \notin \mathcal{E}$. Let z_i be the time at which state i is first reached and let k_{ii} count the number of self-transitions on state i. Then,

$$\pi^\star_{T,i} = \sum_{t=0}^{T} \mathbf{P}[z_i = t]\mathbf{P}[k_{ii} \geq T - t]. \tag{2}$$

Starting with $\mathbf{P}[k_{ii} \geq 0] = 1$ and $\mathbf{P}[z_i = 0] = \pi_{0i}$, we can compute the required quantities recursively as

$$\mathbf{P}[k_{ii} \geq t] = \mathbf{P}[k_{ii} \geq t - 1] \cdot \frac{\eta_{ii} + t - 1}{\sum_\ell \eta i\ell + t - 1}, \tag{3}$$

$$\mathbf{P}[z_i = t] = \sum_j \sum_{t'=1}^{t-1} \Bigg[\mathbf{P}[z_j = t'] \cdot \mathbf{P}[k_{jj} \geq t - t'] \cdot \frac{\eta_{ji}}{\sum_\ell \eta_{j\ell} + (t - t')} \Bigg], \tag{4}$$

for $t = 1, \ldots, T$.

This explicit decomposition of the predictive state distribution leads to a proof of Proposition 1, which we briefly recall here.

Proposition 1. *Let $(\boldsymbol{A}, \boldsymbol{B})$ be any generalized Dirichlet mixture of Markov chains on a graph $\mathcal{G} = ([N], \mathcal{E})$, and let $\boldsymbol{\pi}_0$ be an initial state distribution. If $\mathcal{G}$ has no cycle of length greater than one, then $\boldsymbol{\pi}^\star_T$ can be computed exactly in time $O(T^2N^2)$.*

Proof. The predictive state distribution $\boldsymbol{\pi}^\star_T$ can be computed exactly using (2), (3) and (4). There are $N \cdot T$ distinct quantities to compute for (3), each with running time $O(1)$. Similarly, there are $N \cdot T$ distinct quantities to compute for (4), each with running time $O(NT)$. Finally, (2) involves N distinct quantities with running time $O(T)$ each. Adding up the contributions, the total running time is $O(NT \cdot 1 + NT \cdot NT + N \cdot T) = O(T^2N^2)$. □

B.2 EXACT ALGORITHM FOR THE GENERAL CASE

Given a discrete-time mixture model, an initial distribution $\boldsymbol{\pi}_0$ and a time horizon T, we seek to predict the marginal state distribution after T steps, $\boldsymbol{\pi}^\star_T$. A naive solution involves of enumerating all paths of length T, with running time exponential in T. We now introduce an alternative procedure that computes $\boldsymbol{\pi}^\star_T$ exactly with running time polynomial in T.

Let $\boldsymbol{K}_t \in \mathbf{N}^{N \times N}$ be a matrix counting the number of times each transition has occurred up to time t. We write

$$\boldsymbol{\pi}^\star_T = \sum_{\boldsymbol{K}} \mathbf{P}[s_T = i, \boldsymbol{K}_T = \boldsymbol{K}],$$

where $\boldsymbol{K}$ ranges over all integer-valued matrices whose entries sum up to T. Starting from

$$\mathbf{P}[s_0 = i, \boldsymbol{K}_0 = \mathbf{0}_{N \times N}] = \pi_{0i},$$

we can recursively compute

$$\begin{aligned}
\mathbf{P}[s_t = i, \boldsymbol{K}_t = \boldsymbol{K}] &= \sum_j \mathbf{P}[s_t = i, s_{t-1} = j, \boldsymbol{K}_t = \boldsymbol{K}] \\
&= \sum_j \mathbf{P}[s_t = i, s_{t-1} = j, \boldsymbol{K}_{t-1} = \boldsymbol{K} - \boldsymbol{\Delta}^{ji}] \\
&= \sum_j \Big(\mathbf{P}[s_t = i \mid s_{t-1} = j, \boldsymbol{K}_{t-1} = \boldsymbol{K} - \boldsymbol{\Delta}^{ji}] \\
&\qquad \cdot \mathbf{P}[s_{t-1} = j, \boldsymbol{K}_{t-1} = \boldsymbol{K} - \boldsymbol{\Delta}^{ji}] \Big)
\end{aligned}$$

for $t = 1, \ldots, T$, where $\boldsymbol{\Delta}^{ij}$ is the $N \times N$ indicator matrix whose entry (i, j) is 1 and all other entries are 0, and where

$$\begin{aligned}
&\mathbf{P}[s_t = i \mid s_{t-1} = j, \boldsymbol{K}_{t-1} = \boldsymbol{K}] \\
&\quad = \left(\frac{\alpha_{ji} + k_{ji}}{\alpha_{ji} + \beta_{ji} + \sum_{o \geq i} k_{jo}} \right)^{\mathbf{1}\{i \neq N\}} \\
&\qquad \cdot \prod_{\ell=1}^{i-1} \frac{\beta_{j\ell} + \sum_{o > \ell} k_{jo}}{\alpha_{j\ell} + \beta_{j\ell} + \sum_{o \geq \ell} k_{jo}}.
\end{aligned}$$

In the case of the standard Dirichlet distribution (see Appendix A), the transition probability simplifies to

$$\mathbf{P}[s_t = i \mid s_{t-1} = j, \boldsymbol{K}_{t-1} = \boldsymbol{K}] = \frac{\eta_{ji} + k_{ji}}{\sum_\ell (\eta_{j\ell} + k_{j\ell})}.$$

Running-Time Analysis. The stars and bars theorem implies that $\boldsymbol{K}_t$ can take $\binom{t+N^2-1}{N^2-1}$ different values [Feller, 1968]. Thus, the total number of subproblems we need to solve is given by

$$\sum_{t=0}^{T} \binom{t+N^2-1}{N^2-1} = \binom{T+N^2}{N^2} = O(T^{N^2}),$$

where the first equality follows from the hockey-stick identity [Jones, 1996], a special case of the Vandermonde identity. Each subproblem involves a sum over N terms, leading to an overall running time $O(NT^{N^2})$. In the case where the admissible transitions are restricted to the graph $\mathcal{G} = ([N], \mathcal{E})$, a similar development shows that the running time reduces to $O(d_{\text{avg}} T^{|\mathcal{E}|})$, where d_{avg} is the average node degree. Even though this procedure is more efficient than enumerating all paths of length T, it remains impractical for all but the smallest problems.

B.3 CONVERGENCE OF ALGORITHM 1

We start by proving Proposition 2 in the main text, which we recall here for convenience.

Proposition 2. *For any $\boldsymbol{A}, \boldsymbol{B}$, horizon T, and initial distribution $\boldsymbol{\pi}_0$, let $\hat{\boldsymbol{\pi}}_T$ be the output of Algorithm 1. Then, for any $\epsilon, \delta > 0$, we have*

$$\mathbf{P}[\|\hat{\boldsymbol{\pi}}_T - \boldsymbol{\pi}_T^\star\| < \varepsilon] > 1 - \delta,$$

as long as $L > \frac{11}{\varepsilon^2} \log \frac{N+1}{\delta}$.

Proof. The result follows from the matrix Bernstein inequality [Tropp, 2015, Thm. 1.6.2] applied to the random vectors $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_L\}$, where $\boldsymbol{z}_\ell = \boldsymbol{\pi}_{\ell,T} - \boldsymbol{\pi}_T^\star$. By construction, $\{\boldsymbol{z}_\ell\}$ are jointly independent and $\mathbf{E}[\boldsymbol{z}_\ell] = 0$ for all ℓ. Furthermore, since $\boldsymbol{z}_\ell$ is a difference of two probability vectors, $\|\boldsymbol{z}_\ell\| \leq 2$ for all ℓ and $\|\sum_\ell \boldsymbol{z}_\ell^\top \boldsymbol{z}_\ell\| = \|\sum_\ell \boldsymbol{z}_\ell \boldsymbol{z}_\ell^\top\| \leq 4L$. As a consequence, the matrix Bernstein inequality yields

$$\mathbf{P}\Big[\Big\|\sum_\ell \boldsymbol{z}_\ell\Big\| \geq L\varepsilon\Big] \leq (N+1) \cdot \exp\left(-\frac{L^2\varepsilon^2/2}{4L + 2L\varepsilon/3}\right),$$

and with some basic algebraic manipulations, we obtain the result as formulated in the proposition. □

Note that Proposition 2 holds for any algorithm that averages independent samples centered around the true state distribution. However, we intuitively expect that, for a given budget of samples L, Algorithm 1 returns a better estimate than one obtained by naively sampling entire trajectories. This is because Algorithm 1 first samples from the mixture distribution, and then averages over *all* possible paths, instead of sampling a single path. We verify this empirically in the next section.

Continuous-Time Algorithm. For completeness, we briefly review the continuous-time variant of the sampling procedure introduced in Section 4 of the main text. We present the procedure in Algorithm 2. The predictive state distribution of a CTMC sampled from the mixture distribution is computed on line 3. In practice, the matrix exponential often cannot be computed exactly, but it can be approximated effectively [Al-Mohy and Higham, 2010]. Most numerical libraries and machine-learning frameworks provide the matrix exponential as a primitive.[1]

Algorithm 2 Predictive state distribution.

Require: $\boldsymbol{A}, \boldsymbol{B}$, horizon T, init. dist. $\boldsymbol{\pi}_0$, num. samples L

1: **for** $\ell = 1, \ldots, L$ **do**
2: $\quad \boldsymbol{\Lambda} \leftarrow$ sample from $\prod_{i \neq j} \Gamma(\boldsymbol{\lambda}_{ij} \mid \boldsymbol{\alpha}_{ij}, \boldsymbol{\beta}_{ij})$
3: $\quad \boldsymbol{\pi}_{\ell,T} \leftarrow \boldsymbol{\pi}_0^\top e^{T\boldsymbol{\Lambda}}$
4: **return** $\hat{\boldsymbol{\pi}}_T = \frac{1}{L} \sum_\ell \boldsymbol{\pi}_{\ell,T}$

B.4 EMPIRICAL CONVERGENCE OF ALGORITHM 1

A practical approach to computing the predictive state distribution for *any* model is to sample a small set of trajectories and estimate the distribution empirically by using the samples. We refer to this as the *naive sampling* scheme. In this section, we compare Algorithm 1 to the naive scheme in terms of the quality of the estimated distribution $\hat{\boldsymbol{\pi}}_T$, for a given budget of samples L.

We generate a synthetic problem instance as follows. Setting the number of states to $N = 5$, we sample a matrix $\boldsymbol{H} \in [0, \rho]^{N \times N}$ uniformly at random, for $\rho \in \{1, 10, 100\}$. We interpret this matrix as the parameters of a product of N Dirichlet distributions, a special case of the GDir distribution (see Appendix A). Informally, the larger ρ is, the more the mixture distribution is concentrated around a single DTMC transition matrix. We then sample an initial state i_0 uniformly at random, let $\boldsymbol{\pi}_0 = [\mathbf{1}_{i=i_0}]$, and we estimate the predictive state distribution at horizon $T = 10$. Even though we report results only on a specific experimental setting, our findings appear to be robust to different choices of N, T, and $\boldsymbol{\pi}_0$. We compare the empirical estimate obtained by using $L = 1, ..., 10^3$ samples (collected through naive sampling or Algorithm 1) to the ground truth by computing the ℓ_2-norm of the difference vector. For each value of ρ, we average the performance obtained on $M = 20$ instances and present the results in Figure 1.

In all cases, the ℓ_2 distance to the ground truth appears to decrease as $1/\sqrt{L}$. This is expected: both our proposed approach and naive sampling rely on averaging independent samples centered around $\boldsymbol{\pi}_T^\star$. However, we observe that

[1] For example, `scipy.linalg.expm` in SciPy and `tf.linalg.expm` in TensorFlow.

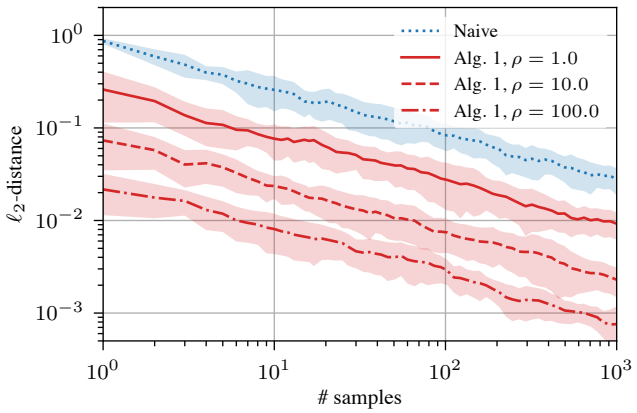

Figure 1: Mean $\pm$ std. (20 instances) of the distance $\|\hat{\boldsymbol{\pi}}_T - \boldsymbol{\pi}_T^\star\|$ as a function of L for the naive sampling scheme and Algorithm 1. For naive sampling, the performance is nearly identical for every value of ρ and we thus draw a single line.

Algorithm 1, which samples from the mixing distribution but then averages over all paths, results in samples with lower variance. This, in turn, leads to better estimates for any given sampling budget L. In fact, Algorithm 1 requires 10–1000$\times$ fewer samples than naive sampling in order to reach a given accuracy. The gains depend on the shape of the mixture distributions; For $\rho = 1$ (strongly multimodal mixture distribution) the advantage is relatively modest, whereas for larger values of ρ it becomes important.

C EXPERIMENTAL EVALUATION

In this Appendix, we provide more details on the datasets and baselines presented in the main text. In addition, we provide (as part of the supplementary material) an archive that contains a small software library written in the Python language and computational notebooks (using the Jupyter Notebook format) that enable reproducing the experiments of Sections 5 and B.4 from raw data.

C.1 DATASETS

We provide additional information on the datasets studied in Section 5 of the main text. Summary statistic including the number of states N, the number of admissible transitions $|\mathcal{E}|$ and the number of sequences M is provided in Table 1.

SLEEP. The dataset is studied by Kneib and Hennerfeind [2008] and is available on Thomas Kneib's webpage.[2] Each sequence captures the sleep patterns of an individual. There are three states representing rapid eye-movement (REM) sleep, non-REM sleep, and awake.

[2]See: `https://www.uni-goettingen.de/de/551628.html`.

VENTICU. The dataset is studied by Grundmann et al. [2005] and is available on Richard J. Cook's webpage.[2] Each sequence represents a patient in an intensive care unit. The four states capture ventilation (on and off), discharge, and death, respectively.

EBMT. The dataset is studied by Fiocco et al. [2008] and is available on Richard J. Cook's webpage.[2] Each sequence captures patient outcomes after blood and marrow transplantation. The six states represent outcomes such as remission, adverse events, relapse, death, and combinations thereof.

CUSTOMERS. This dataset is not available publicly at this time. Each sequence represents a customer and their relationship to a business over time. The three states represent: using the free service, subscribing to the paid service, and not using the service, respectively.

C.2 FINITE MIXTURES OF MARKOV CHAINS

In order to train finite mixture models, we follow Cadez et al. [2003]. We stop the EM algorithm as soon as the log-likelihood increases by less than 0.1% during one iteration. In order to select the number of mixture components, we perform a search over $L \in \{2, 3, 5, 10, 20\}$. We report the results corresponding to the value of L which minimizes the log-likelihood on the hold-out set.

C.3 COMPUTATIONAL SETUP

Our experiments are run on a Google cloud `n1-standard-32` instance with 32 vCPUs and 120 GB RAM. Our code relies on the following versions of popular Python packages:

- `jax==0.2.13`
- `jaxlib==0.1.67`
- `numpy==1.19.5`
- `scipy==1.6.3`

Bibliography

A. H. Al-Mohy and N. J. Higham. A new scaling and squaring algorithm for the matrix exponential. *SIAM Journal on Matrix Analysis and Applications*, 31(3):970–989, 2010.

I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery*, 7:399–424, 2003.

R. J. Connor and J. E. Mosimann. Concepts of independence for proportions with a generalization of the Dirichlet dis-

Table 1: Summary statistics of the four datasets.

Dataset	Reference	Type	N	$\lvert\mathcal{E}\rvert$	M
SLEEP	Kneib and Hennerfeind [2008]	Continuous	3	6	70
VENTICU	Grundmann et al. [2005]	Continuous	4	6	747
EBMT	Fiocco et al. [2008]	Continuous	6	12	2279
CUSTOMERS	—	Discrete	3	9	144 510

tribution. *Journal of the American Statistical Association*, 64(325):194–206, 1969.

W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, third edition, 1968.

M. Fiocco, H. Putter, and H. C. van Houwelingen. Reduced-rank proportional hazards regression and simulation-based prediction for multi-state models. *Statistics in Medicine*, 27(21):4340–4358, 2008.

H. Grundmann, S. Bärwolff, A. Tami, M. Behnke, F. Schwab, C. Geffers, E. Halle, U. B. Göbel, R. Schiller, D. Jonas, et al. How many infections are caused by patient-to-patient transmission in intensive care units? *Critical Care Medicine*, 33(5):946–951, 2005.

C. H. Jones. Generalized hockey stick identities and N-dimensional block walking. *Fibonacci Quarterly*, 34(3): 280–288, 1996.

T. Kneib and A. Hennerfeind. Bayesian semiparametric multi-state models. *Statistical Modelling*, 8(2):169–198, 2008.

J. R. Norris. *Markov Chains*. Cambridge University Press, 1998.

J. A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.