Recovering Markov Models from Closed-Loop Data

Jonathan Epperlein†, Robert Shorten‡, and Sergiy Zhuk†

Abstract. Situations in which recommender systems are used to augment decision making are becoming prevalent in many application domains. Almost always, these prediction tools (recommenders) are created with a view to affecting behavioural change. Clearly, successful applications actuating behavioural change, affect the original model underpinning the predictor, leading to an inconsistency. This feedback loop is often not considered in standard so-called Big Data learning techniques which rely upon machine learning/statistical learning machinery. The objective of this paper is to develop tools that recover unbiased user models in the presence of recommenders. More specifically, we assume that we observe a time series which is a trajectory of a Markov chain \( R \) modulated by another Markov chain \( S \), i.e. the transition matrix of \( R \) is unknown and depends on the current state of \( S \). The transition matrix of the latter is also unknown. In other words, at each time instant, \( S \) selects a transition matrix for \( R \) within a given set which consists of known and unknown matrices. The state of \( S \), in turn, depends on the current state of \( R \) thus introducing a feedback loop. We propose an Expectation-Maximization (EM) type algorithm, which estimates the transition matrices of \( S \) and \( R \). Experimental results are given to demonstrate the efficacy of the approach.

Key words. Markov chains; identification; feedback; Expectation-Maximization; recommenders;

AMS subject classifications. 60J10; 90B20; 60J20; 93E12; 93E35

1. Introduction. Our starting point for this paper is a frequently encountered problem that arises in the area of Smart Cities research. Many decision support systems that are designed to solve Smart City problems are data-driven: that is data, sometimes in real time, are used to build models to drive the design of recommender systems. Almost always, these datasets have been treated as if they were obtained in the open-loop setting. However, this is rarely the case and frequently the effects of recommenders are inherent in datasets used for model building. This creates new challenges for the design of feedback systems in cities.

We shall consider systems with the following structure: the process, which generates the data; the model, which represents the behaviour of the process; and the decision support tool, which intermittently influences the process. In our setup, data from the process is used to build the model. Typically, the model is used to construct a decision support tool which itself then influences the process directly. This creates a feedback loop in which the process, decision support tool and the model are interconnected in a complicated manner. As a result, the effect of the decision support tool is to bias the data being generated by the process, and consequently to bias any model that is constructed naïvely from the data. To be a little more specific we now illustrate such effects by means of the following application that we are developing in the context of our automotive research.

Consider a driver who drives a car regularly. In order to design a recommender system for this driver we would like to build a model of his/her behaviour. For example, in order to warn the driver of, say, roadworks, along a likely route, we might use this model to predict the route of the driver. A schematic of the proposed in-car architecture is depicted in Figure 1. The recommender uses a model of driver behaviour to issue intermittent recommendations. Observations of driver behaviour are then used to build a refined driver model which in turn is used as an input to the recommender system. Clearly, the effect of the recommender is to bias the driver model over time, thus eventually
rendering the latter ineffective as an input to the recommender. The problems are exacerbated in many practical systems due to the presence of several unknown third-party recommender systems (Google Maps, Siri etc.), and by the fact that the driver model may operate from birth-to-death\(^2\) in closed loop. This later fact makes it difficult, or impossible, to even estimate an initial model of driver behaviour. Clearly, in such applications it is absolutely necessary to develop techniques that extract the behaviour of the driver while under the influence of the feedback from a number of recommender systems. The results presented in this paper represent our first small step in this direction.

1.1. General comments on related research directions. Dealing with bias arising from closed-loop behaviour is a problem that has arisen in several application domains. In fact, in control theory, the related topic of closed-loop identification is considered to be a very mature area [14, 18, 24]. Roughly speaking, this topic is concerned with building models of dynamic systems while they are being regulated by a controller. A related scenario arises in some adaptive systems when the controller itself is being adjusted on the basis of the dynamical systems model. As in our example, the controller action will bias the estimation of the model parameters. While many established techniques in control theory exist for dealing with such effects; these typically exploit known properties of the process noise and an assumed model structure to un-bias the estimates. Typically, structures such as ARMAX models are assumed to capture the nature of the system dynamics. Recent work on intermittent feedback [15] is also related in spirit to these approaches where control design techniques to deal with feedback loops that are intermittently broken are developed. Before proceeding it is also worth noting that work which is closely related to the control theoretic approaches has also been explored in the economics literature [1, 19, 25, 27]. It is also worth mentioning that biasing effects due to feedback, can, in some circumstances, be dealt with in a mixture model context and treated as a Gaussian separation problem [17, 26].

More recently, several authors in the context of Smart Cities and Big data [6, 23], have realised that closed-loop effects represent a fundamental challenge in the design of recommender systems. In [6] the authors discuss the inherent closed-loop nature of data-sets in cities, and in [5, 23] explicitly discuss the influence of feedback on the fidelity of recommender systems. As an example of a specific result, [23] present an empirical technique to recover user rankings in the presence of a recommender under an assumed interaction model between user and recommender. The performance degradation of online recommender systems due to ignoring the feedback effects when the standard learning

---

\(^2\) by that we mean that the driver always operates under the potential influence of a recommender. Thus, given any observation, we do not know whether the recommender is acting, or the driver.
mechanism (e.g. train with user feedbacks, make recommendations, collect user feedbacks, re-train) is used, was also reported in [16].

Roughly speaking, the approach outlined in the first paragraph of Section 1.1 assumes deterministic models subject to stochastic noise, whereas the approaches mentioned in the second paragraph of Section 1.1 assume non-deterministic models. The work reported here is more closely aligned with the latter line of enquiry. Specifically, in this paper we are interested in reconstructing Markov models that are operating under the influence of a recommender. To this end we assume that (i) recommenders and users can both be modelled by Markov chains, and (ii) recommendations are either accepted fully or have no influence at all, i.e. observed decisions are always unbiased - either a recommender or the driver.

This work is most related to [7] with the important distinction of closed-loop modulation, i.e. the modulated visible process is allowed to in turn modulate the the modulating latent process. Note that the Markovian assumption of user and recommender behaviour is convenient for many applications: for example, in the automotive domains [9, 11]. More specific technical comments to place our work in the context of reconstructing Markov models from the data are given below in Section 5.

1.2. Preliminaries. Notation. To compactly represent discrete state spaces we write \([N] := \{1, \ldots, N\}\). For a function \(M : [N] \to \mathbb{R}^{m \times n}\) mapping such a discrete finite set to a set of matrices, we refer to each value \(M(k) \in \mathbb{R}^{m \times n}\) as a page of \(M\). Matrices will be denoted by capital letters, their elements by the same letter in lower case, and we denote the set of \(n \times m\) row-stochastic matrices, i.e. matrices with non-negative entries such that every row sums up to 1, by \(\mathcal{M}^{n \times m}\), and \(\mathcal{M}^m := \mathcal{M}^{m \times m}\). If \(A \in \mathcal{M}^m\) then \(\text{Vec}(A) \in \mathbb{R}^{m^2}\) denotes a vector obtained by stacking the columns of \(A\). For compatible matrices, \(M \otimes N\) is the Kronecker product, and \(M \circ N\) denotes the Hadamard (or element-wise) product. A partition \(\Gamma\) of \([N]\) is a set \(\{\Gamma_1, \ldots, \Gamma_p\}\) such that \(\Gamma_i \subseteq [N]\), \(\cup_i \Gamma = [N]\), \(\Gamma_i \cap \Gamma_j = \emptyset\) if \(i \neq j\). Each partition then also defines a membership function \(\gamma : [N] \to [p]\) by \(\gamma(i) := k\) such that \(i \in \Gamma_k\). We write \(P(W = w)\) for the probability of the event that a realisation of the discrete random variable \(W\) equals \(w\), and \(P(W = w \mid V = v)\) the probability of that same event \(W = w\) conditioned on the event \(V = v\). Abusing notation, we shall, where appropriate, also denote random variables by capital letters and their realisations by the same letter in lower case. For convenience we will sometimes write \(P(w \mid v)\) instead of \(P(W = w \mid V = v)\) if there is no risk of ambiguity, and, for a set of parameters \(\mu\) parametrising a probability distribution, \(P(W = w \mid \mu)\) is taken to denote the probability of the event \(W = w\) if the parameters are set to \(\mu\).

Markov chains herein are sequences of random variables \(\{X_t\}_t\) indexed by the time \(t \in \mathbb{N} = \{0, 1, 2, \ldots\}\). The realisation \(x_t \in [N]\) of \(X_t\) is the state of the Markov chain at time \(t\), and \([N]\) is its state space. The probability distribution of \(X_0\) is denoted by \(\pi_0\), and the probability distribution of each following state is given by \(P(X_t = j \mid X_{t-1} = i) = a_{ij}\), and the matrix \(A \in \mathcal{M}^{N}\) with entries \(a_{ij}\) is a row-stochastic transition probability matrix.

2. Problem Statement. As noted above, we assume that the driver and the recommender are Markovian. Given a possibly incomplete description of Markov chains modelling the recommender systems, and no knowledge of when these systems are engaged, our aim is to estimate the probability transition matrix of the Markov chain representing the driver, and the levels of engagement of each recommender, using only observed data. In what follows we formalize this setup, and give an Expectation-Maximization (EM) algorithm to estimate the parameters of the unknown driver model.

2.1. “Open-Loop” Markov-Modulated Markov Chains. Consider a Markov chain \(R\) with state space \([R]\) and state \(r_t \in [R]\), in which the transition probabilities

\[
P(R_t = j \mid R_{t-1} = i, S_{t-1} = s) = a_{ij}^{R}(s)
\]

depend on a latent random variable \(S_t\). We can say that the Markov chain is modulated by the random variable \(S_t\), and if \(S_t\) is itself the state of another Markov chain \(S\) with transition matrix
$A^S$ and state space $[S]$, then we are dealing with a \textit{Markov-modulated Markov chain} (MMMC); Markov modulation is an established model in the literature on inhomogeneous stochastic processes, see e.g. [7].

Formally, the MMMC is defined by the tuple $\mu = (\pi^R, \pi^S, A^R(\cdot), A^S)$, where $A^R : [S] \to \mathcal{M}^R$ and $\pi^R, \pi^S$ denote the distributions of $S_0$ and $R_0$, respectively. That means for instance that if $S$ only has a single state $s$, $\mu$ is a regular Markov chain with transition matrix $A^R(s)$ and initial probability $\pi^R$. We assume that we observe the state of $R$, but not the state of $S$.

Because the transition probabilities in the latent Markov chain $S$ do not depend on the state of the visible chain $R$, we refer to $\mu$ as an \textit{open-loop} MMMC to distinguish it from what follows. This models the case when the switching between the transition matrices $A^R$ occurs independently of the current state $r_t$ of $R$.

The joint process of $Q_t = (S_t, R_t)$ has transition probabilities

\begin{align*}
P(Q_t = (s', r') \mid Q_{t-1} = (s, r)) &= P(R_t = r' \mid R_{t-1} = r, S_{t-1} = s, S_t = s') \times P(S_t = s' \mid R_{t-1} = r, S_{t-1} = s) = a^R_{r', r}(s) a^S_{ss'},
\end{align*}

where the first cancellation means that the decision at time $t-1$ is not influenced by the state of the modulating random variable at time $t$, and the second cancellation follows from the open-loop assumption, i.e. that the modulating Markov chain $S$ evolves independently of $R$. The estimation of $S$ and $R$ for the case of continuous time open-loop MMMC has been discussed in [7].

**Remark:** We are dealing with the case when the data consists of a finite time series of observations of a single trajectory $(r_0 r_1 \cdots r_T)$ of the Markov chain $R$ and no (estimate of) the distribution of $R_t$ is available. While if the distributions are available, standard methods of state-space identification apply, here the estimation of the parameters of $\mu$ requires statistical methods such as maximum likelihood estimation.

2.2. Closed-Loop Markov-Modulated Markov Chains. As a generalisation, we consider the case where $S$ is dependent on the state of $R$: that is, the probabilities of transitioning from one state $s$ to another state $s'$ then depend on what the current state $r_t$ is. We will be referring to this as an \textit{closed-loop} MMMMC or clMMMC for short. A clMMMC can be used to model that one transition matrix $A^R(s)$ might be more likely to be switched to in some regions of the visible state space $[R]$, or that switching can only occur when the system is in specific configurations. This is exactly the situation which arises in our automotive example, see Section 4.2.

Formally, we now also allow for the latent Markov chain $S$ to be modulated by the current state of the visible chain $R$. To keep the developments general, assume that – instead of one page of $A^S$ corresponding to each state of $R$ – there is a partition $\Gamma = \{\Gamma_1, \ldots, \Gamma_p\}$ of $[R]$ such that there is a page in $A^S$ for each $\Gamma_i$. Hence, we now have $A^S : [p] \to \mathcal{M}^S$, with

\begin{align*}
P(S_t = j \mid S_{t-1} = i, R_{t-1} = r) = a^S_{ij}(\gamma(r)).
\end{align*}

The open-loop case then corresponds to $\Gamma = \{[R]\}$ (i.e. $p = 1$ and $\gamma(i) \equiv 1$) and the joint process $Q_t = (S_t, R_t)$ has transition probabilities (compare to the open-loop formula above):

\begin{align*}
P(Q_t = (s', r') \mid Q_{t-1} = (s, r)) &= P(R_t = r' \mid R_{t-1} = r, S_{t-1} = s) \times P(S_t = s' \mid R_{t-1} = r, S_{t-1} = s) = a^R_{r', r}(s) a^S_{ss'}(\gamma(r)).
\end{align*}

Such a clMMMC is also represented by a tuple $\mu = (\pi^R, \pi^S, A^R(\cdot), A^S(\cdot); \Gamma)$, only that now, $A^S(\cdot)$ has pages, too.

\footnote{Recall from Section 1.2 that $\gamma(\cdot)$ is the membership function of $\Gamma$.}
3. Likelihood and Parameter Estimation. In this section we develop an iterative algorithm to estimate the parameters of a clMMMC \( \mu = (\pi^R, \pi^S, A^S(\cdot), A^R(\cdot); \Gamma) \) given a sequence of observations \( (r_0 r_1 \cdots r_T) \), a partition \( \Gamma = \{ \Gamma_1, \ldots, \Gamma_p \} \) of \([R]\) and the size \( S \) of the state space of \( S \). The derivation is close in spirit to the classical Baum-Welch (BW) algorithm (see e.g. [21] and the numerous references therein): our algorithm maximises at every iteration a lower bound on the likelihood improvement, and gives rise to re-estimation formulae (15) that utilise forward and backward variables. These look similar to, but differ in subtle ways from, the ones for the BW algorithm. The relationship with HMMs is further discussed in Section 5.

3.1. Likelihood of \( \mu \), forward- and backward variables. Since the estimate to be obtained is a maximum likelihood (ML) estimate, the efficient computation of the likelihood of a given clMMMC \( \mu \) plays a central role in what follows. For a given \( \mu \), the joint probability of sequences \( (r_0 r_1 \cdots r_T) \) and \( (s_0 s_1 \cdots s_T) \) being the trajectory of the visible Markov chain \( R \) (resp. latent Markov chain \( S \)) is

\[
P(s_0, \ldots, s_T, r_0, \ldots, r_T \mid \mu) = \pi^{R}_0 \pi^{S}_0 \prod_{t=1}^{T} P(s_t, r_t \mid s_{t-1}, r_{t-1}, \mu) \\
= \pi^{R}_0 \pi^{S}_0 \prod_{t=1}^{T} \alpha^{S}_{s_{t-1}s_t}(\gamma(r_{t-1})) \alpha^{R}_{r_{t-1}r_t}(s_{t-1})
\]

where the last equality follows by (2). This allows us to compute the probability of observing a sequence \( (r_0 r_1 \cdots r_T) \) given \( \mu \) as follows:

\[
P(r_0 r_1 \cdots r_T \mid \mu) = \sum_{s_0 \in [S] \cdots s_T \in [S]} P(s_0, \ldots, s_T, r_0, \ldots, r_T \mid \mu) \\
= \pi^{R}_0 \sum_{s_0 \cdots s_T} \prod_{t=1}^{T} \alpha^{S}_{s_{t-1}s_t}(\gamma(r_{t-1})) \alpha^{R}_{r_{t-1}r_t}(s_{t-1}) =: \ell(\mu)
\]

where \( \mu \mapsto \ell(\mu) \) is the likelihood of the model \( \mu \). Computation using this direct expression requires, on the order of, \( 2 \times T \times S^T \) operations, and is hence not feasible for large \( T \). Instead, we define the forward variable \( \alpha_t \) with \( S \) elements

\[
\alpha_t(i) := P(S_t = i, R_0 = r_0, \ldots, R_t = r_t \mid \mu)
\]

which can be computed iteratively as follows: \( \alpha_0(j) = \pi^S_j \pi^R_{r_0} \) and

\[
\alpha_t(j) = \sum_{i=1}^{S} \alpha_{t-1}(i) \alpha^S_{j,i}(\gamma(r_{t-1})) \alpha^R_{r_{t-1}r_t}(i), \quad j = 1, \ldots, S
\]

or, in matrix form: \( \alpha_0 = \pi^S \pi^R_{r_0} \) and

\[
\alpha_t = (A^S(\gamma(r_{t-1})))^T \left( a^R_{r_{t-1}r_t}(\cdot) \circ \alpha_{t-1} \right),
\]

where the notation \( a^R_{r_{t-1}r_t}(\cdot) \) means a column vector of the \( (r_{t-1}, r_t) \)-elements of the matrix \( A^R(k) \) as \( k \) runs from 1 to \( S \). \(^4\)

\[
\ell(\mu) = \sum_{i=1}^{S} P(S_T = i, r_0, \ldots, r_T \mid \mu) = \sum_{i} \alpha_T(i) = 1^T A_T
\]

\(^4\)Very much analogous to Matlab’s colon notation, or slicing in numpy.
can be computed with on the order of $TS^2$ computations.

An analogous concept that will be required later is the *backwards variable*

\begin{equation}
\beta_t(i) := P(r_{t+1}, \ldots, r_T | S_t = i, r_t),
\end{equation}

which can also be computed via the iteration $\beta_T(j) = 1$, $\beta_{t-1}(j) = \sum_i \beta_t(i) a^R_{r_{t-1}r_t}(j) a^S_{ij}(\gamma(r_{t-1}))$, or in matrix form: $\beta_T = 1$ and

\begin{equation}
\beta_{t-1} = (A^S(\gamma(r_{t-1})) \beta_t) \circ a^R_{r_{t-1}r_t}(\cdot).
\end{equation}

### 3.2. Auxiliary Function $Q(\mu, \mu')$

We can identify a cMMMC $\mu = (\pi^R, \pi^S, A^R(\cdot), A^S(\cdot); \Gamma)$ (in a slight abuse of notation) with the $(R + S + SR^2 + pS^2)$-dimensional vector $\mu = \begin{bmatrix} \text{Vec}(\Omega_R) \\ \text{Vec}(\Omega_S) \end{bmatrix}$, where

\begin{align*}
\Omega_R^T &= \begin{bmatrix} \pi^R \top \\ A^R(1) \\ \vdots \\ A^R(S) \end{bmatrix}, & \Omega_S^T &= \begin{bmatrix} \pi^S \top \\ A^S(1) \\ \vdots \\ A^S(p) \end{bmatrix}.
\end{align*}

The **convex** set of all cMMMCs is then given by

\[ \Lambda := \left\{ \mu = \begin{bmatrix} \text{Vec}(\Omega_R) \\ \text{Vec}(\Omega_S) \end{bmatrix} : \Omega_R^T \in \mathcal{M}^{1 + RS; R}, \Omega_S^T \in \mathcal{M}^{Sp+1; S} \right\}. \]

Let $\Lambda^+ := \{ \mu \in \Lambda : \mu_i > 0 \}$. Following [2], we define the auxiliary function $Q(\mu, \mu')$ of $\mu, \mu' \in \Lambda$ by

\begin{equation}
Q(\mu, \mu') := \sum_{s_0, \ldots, s_T} P(s_0, \ldots, s_T, r_0, \ldots, r_T | \mu) \cdot \log P(s_0, \ldots, s_T, r_0, \ldots, r_T | \mu'),
\end{equation}

where $s_0, \ldots, s_T$ run through the $S^T$ possible sequences of the latent state $S_t$. If parameters $\mu'_i$ are zero where $\mu_i > 0$, then we can have the case $P(s_0, \ldots, r_T | \mu') = 0$ and $P(s_0, \ldots, r_T | \mu) > 0$; in this case $Q(\mu, \mu') := -\infty$. If $P(s_0, \ldots, r_T | \mu) = P(s_0, \ldots, r_T | \mu) = 0$, we set $Q(\mu, \mu') = 0$.

The following lemma establishes a representation for $Q$ in terms of the elements of $\mu' = (\pi^R, \pi^S, A^S, A^R; \Gamma)$:

**Lemma 1.** The function $Q(\mu, \mu')$ can be rewritten as

\begin{equation}
Q(\mu, \mu') = \log \pi^R_{r_0} \ell(\mu) + \sum_{i=1}^{S} \log \pi^S_i \sum_{j=1}^{S} \xi_1(i, j)
\end{equation}

\begin{equation}
+ \sum_{i, j, t=1}^{S, \ldots, T} \left[ \log \left(a^S_{ij}(\gamma(r_{t-1})) \right) + \log \left(a^R_{r_{t-1}r_t}(i) \right) \right] \xi_1(i, j)
\end{equation}

where

\begin{equation}
\xi_t(i, j) := P(S_{t-1} = i, S_t = j, r_0, \ldots, r_T | \mu)
\end{equation}

can be computed as follows:

\begin{equation}
\xi_t(i, j) = a_{t-1}(i) a^R_{r_{t-1}r_t}(i) a^S_{ij}(\gamma(r_{t-1})) \beta_t(j),
\end{equation}

and the variables carrying a \textbullet constitute $\mu'$.

Additionally, an application of Jensen’s inequality yields:

**Lemma 2.** The improvement in log-likelihood satisfies the lower bound

\begin{equation}
\ell(\mu) \left( \log \ell(\mu') - \log \ell(\mu) \right) \geq Q(\mu, \mu') - Q(\mu, \mu).
\end{equation}

\(^5\)Which amounts to setting $0 \log(0) = 0$. 

3.3. EM-Algorithm for Parameter Estimation. The algorithm proceeds by maximising the lower bound on the log-likelihood improvement set forth in (14) at every iteration.

It should be clear from (11) that the best estimate of \( \pi^R \) is the \( r_0 \)-th canonical Euclidian basis vector \( e_{r_0} \). The remaining parameters of \( \mu \) can be iteratively estimated by repeatedly applying

**Theorem 3.** The unique maximiser \( \mu^* = M(\mu) \) of \( Q(\mu, \cdot) \) is given by

\[
\begin{align}
\pi^{S'}_i &= \frac{\sum_{j=1}^S \xi_t(i, j)}{\sum_{j=1}^T \alpha_T} \\
a^{ij}_l(t) &= \frac{\sum_{t: \gamma_t(l)=l} \xi_t(i, j)}{\sum_{k=1}^S \sum_{t: \gamma_t(l)=l} \xi_t(i, k)} \\
a^{R^*}_{mn}(i) &= \frac{\sum_{k=1}^R \sum_{t: \gamma_t(l)=l} \xi_t(i, k)}{\sum_{m=1}^S \sum_{n=1}^R \sum_{k=1}^{\gamma_t(l)=l} \xi_t(i, k)},
\end{align}
\]

where \( i, j = 1, \ldots, S, l = 1, \ldots, p \), and \( m, n = 1, \ldots, R \).

In the \( k \)-th iteration of the EM algorithm, the E-step consists of computing \( \xi_t(\cdot, \cdot) \) for the current estimate \( \mu^k \), the M-step yields an estimate \( \mu^{k+1} = M(\mu^k) \) with improved likelihood. This procedure will, at least in the limit, reach a stationary point\(^6\) of the likelihood as guaranteed by

**Corollary 4.** For any \( \mu_0 \in \Lambda \), a sequence \( \mu^k := M^{(k)}(\mu_0) \), \( M \) defined by (15), has a limiting point \( \mu^\infty \) which is the fixed point of \( M : M(\mu^\infty) = \mu^\infty \). Every fixed point \( \mu^\infty \) is a stationary point of the likelihood maximization problem: \( \ell(\mu) \rightarrow \max_{\mu \in \Lambda}, \ i.e. \)

\[
(\nabla \ell(\mu^\infty))^\top (\mu - \mu^\infty) \leq 0, \quad \forall \mu \in \Lambda.
\]

Multiple (short) trajectories. So far, we have considered the case when the data consists of one, but long trajectory. An implicit assumption was, that every state of \( R \) is reached (transitioned from) at least once; from Equations (15c) and also (15b), we see that otherwise, there could appear terms like \( 0/0 \) due to empty sums in numerator and denominator. And indeed, there is no way of estimating the transition probabilities from a state that is never reached. However, as in the example in Section 4.2, a typical case is that there are many independent trajectories observed, and while not every state is reached in every trajectory, in the union of all trajectories, each state should be reached at least once. If the latter is not the case then the “statistics” we have does not provide any information regarding such “unreached states” and they must either be removed from the state space or designated as terminal states in a pre-processing step. The corresponding EM-algorithm for parameter estimation for the case of multiple short observations is provided in Appendix C.

4. Examples. Here we illustrate the algorithm’s efficacy in three different scenarios: with synthetic data, i.e. data generated from a cMMMC, denoted \( \mu^{\text{true}} \), but assuming no knowledge beyond \( S \) and \( \Gamma \); then with synthetic data and assuming knowledge of the second decision maker (the recommender), i.e. assuming that \( A^R(2) \) is known; and lastly, a toy example of a practical application, estimation of driver behaviour.

4.1. Synthetic data. No parameters known: We generated \( N_e = 100 \) clMMMCs with \( R = 6, S = 2 \) and \( \Gamma = \{ [R] \} \) (i.e. the open-loop case), and a trajectory of length \( T = 5000 \) for each of them. We then ran the described algorithm from randomly chosen initial guesses. The same was repeated for the same \( N_e \) clMMMCs, only that now, \( \Gamma \) was a randomly selected partition of order 2, so that \( p = 2 \) and a second random page was added to \( A^S \).

For the reasons outlined in Section 5.1, we cannot expect to recover (or estimate) the true parameters \( A^R, A^S, \ldots \), but only a cMMMC \( \mu^{\text{est}} \) that parametrises a similar stochastic process over

\(^6\)Not to be confused with the stationary distribution of a Markov chain.
[R]. As a proxy for this, we check that in most cases, \( \mu^{\text{est}} \) achieves at least the same likelihood of having generated the observed trajectory as \( \mu^{\text{true}} \): Computing \( \log \ell(\mu^{\text{est}}) - \log \ell(\mu^{\text{true}}) \) for all \( N_e \) experiments, we find an interquartile range\(^7\) of [30.8 42.2] for open loop and [15.2 164.0] for closed loop. As expected, the parameters of \( \mu^{\text{est}} \) bear little resemblance to those of \( \mu^{\text{true}} \).

\( A^R(2) \) known: For the second example, we again use synthetic data, however this time we help the estimation by assuming that one of the “decision makers” is known. The setup is exactly the same as in the previous example, except that \( R = 8 \) and that we assume \( A^R(2) \) to be known. The modification to the algorithm is trivial: \( A^R(2) \) is simply not re-estimated.

In this case, we recover \( A^S \) and \( A^R(1) \) to high accuracy. “Accuracy” is hereby measured through statistical distances: since the transition matrices of Markov chains consist of probability distributions, absolute or relative matrix norms are not a good measure of distance between Markov chains. Instead, we consider a statistical distance between the estimated and true probability distributions. One of the simplest such distances is the total variation (TV) distance (see e.g. [12, Ch. 4]), which is given by the maximal difference in probability between two distributions. For probability distributions \( f \) and \( g \) over a discrete space \( \Omega \), this is simply

\[
\|f - g\|_{TV} = \max_{A \subseteq \Omega} f(A) - g(A) = \frac{1}{2} \sum_{\omega \in \Omega} |f(\omega) - g(\omega)|.
\]

We consider here two applications of TV distance to Markov chains. The first is to take the TV distance between the stationary distributions, which concretely amounts to considering the subset \( \rho \) of the state space \([R]\) such that \( P(X_t \in \rho \mid A^{R,\text{est}}(1)) - P(X_t \in \rho \mid A^{R,\text{true}}(1)) \) is maximised (for large enough times \( t \) such that the stationary distribution is reached). If we let \( \psi^{\text{true}} \) and \( \psi^{\text{est}} \) denote the stationary distributions, then

\[
\|A^{R,\text{est}}(1) - A^{R,\text{true}}(1)\|_{\text{stat}} := \|\psi^{\text{est}} - \psi^{\text{true}}\|_{TV}.
\]

However, this is a coarse measure: different Markov chains can have equal stationary distributions. Hence, the second metric incorporates the distance between the individual rows by considering the expectation (under the true stationary distribution \( \psi^{\text{true}} \)) of the TV distance between the estimated and the true row; this equals the sum of the distances between the true and estimated transition probabilities from all states \( i \), weighted by the probability of being in state \( i \):

\[
\|A^{R,\text{est}}(1) - A^{R,\text{true}}(1)\|_{\text{exp}} := \sum_i \psi_i^{\text{true}} \|A_i^{R,\text{est}}(1) - A_i^{R,\text{true}}(1)\|_{TV},
\]

where \( M_i \) denotes the \( i \)-th row of matrix \( M \).

The results (for \( A^R(1) \) only, the analysis and results for \( A^S(\cdot) \) are analogous and are hence omitted) are shown in Figure 2, and we see that the distance for both introduced metrics is often below 10%.

4.2. A model of driver behaviour. Recent research, e.g. [9, 11, 22], suggests that Markov-based models are good approximations of driver behaviour and can be used e.g. for route prediction. Here, we illustrate how cMMMC\(s\) can be used to identify a driver’s preferences when some trips are planned by a recommender system, whose preferences are known, while the other trips are planned by the driver.

Specifically, consider the map in the left panel of Figure 3, which depicts a (very small toy) model of a driver’s possible routes from origin “O” to destination “D.” The houses, as an example,
correspond to schools, that should be avoided in the hour before classes start and after classes end for the day, so there is a route past them and one around them. We assume that if a trip falls into that time frame, the recommender takes over and, with known probabilities, routes the driver either past or around each school; these probabilities make up $A_R^{(2)}$. Otherwise, the driver follows his/her preferences, which constitute $A_R^{(1)}$; this is the matrix we would like to estimate. We generated $N_e = 50$ sets of data by simulating $N_t = 200$ trips on the graph shown in the right panel of Figure 3; this is the line graph of the map, where each road segment corresponds to a node, and an edge goes from node $i$ to node $j$ if it is possible to turn into road segment $j$ from $i$. Each trip has a probability of $p_r = 0.3$ to be planned by the recommender. If a trip was planned by the recommender (resp. driver), a trajectory was generated by a Markov chain with transition matrix $A_R^{(2)}$ (resp. $A_R^{(1)}$) originating in node 1 and terminating when returning to node 1.

For estimation in the cMMMC framework, all trips are then concatenated to form one long trajectory and $A_S$ and $A_R^{(1)}$ are estimated for an oMMMC, i.e. for $\Gamma = \{9\}$. $A_R^{est(1)}$ is then an estimate of the driver preferences. The results are shown in the first column of Figure 4 and are satisfactory already; however, we can leverage the closed-loop framework to include the additional knowledge that the decision maker (i.e. the page of $A_R$ used) can only change after a trip is finished. Because the decision which page of $A_R$ to use at time $t$ is made at $t - 1$, see (1), this means we have to allow for the state of $S$ to change on the road segments prior to reaching the destination. We hence let $\Gamma = \{8, 9\}, \{1, \ldots, 7\}$ and $A_S^{(2)} = I_2$. $A_S^{(1)}$ needs to be identified. The results are shown in Figure 4. Additionally, we can interpret $\psi_{S, est}^2$, the second element of the stationary distribution of $A_S^{est(1)}$ as an estimate of $p_r$. For the open-loop case, we obtain $0.352(\pm 0.0583)$, whereas the cMMMC estimation yields $0.295(\pm 0.044)$.
Figure 3. The map of our small toy model and its abstraction as the line graph of the road model. The direction of traffic is from left to right only. The origin and destination are merged into node 1. The weights denote the transition probabilities for the driver and the recommender system. When there is no weight, then the transition probability is 1 for both.

Figure 4. Results from estimating the driver preferences in Section 4.2. Assuming open loop yields acceptable results; adding the information that only full trips are planned by either driver or recommender improves accuracy considerably. Mean and standard deviation are given.

5. Relationship with Hidden Markov Models. As is evident from the similar forms of forward and backwards variables, the auxiliary function $Q$ and the re-estimation formulas (15), there is a close relationship between closed-loop Markov modulated Markov chains and Hidden Markov Models (HMMs). Formally:

Proposition 5. The clMMMC $\mu = (\pi^R, \pi^S, A^R(\cdot), A^S(\cdot); \Gamma)$ defines the same visible process $\{R_t\}$
as the Hidden Markov Model \( \lambda = (\pi, W, B) \), with

\[
\pi = \pi^S \otimes \pi^R = \begin{bmatrix}
\pi_1^S \pi_1^R \\
\pi_1^S \pi_2^R \\
\vdots \\
\pi_S^S \pi_R^R
\end{bmatrix}, \quad B = 1_S \otimes I_R = \begin{bmatrix}
I_R \\
\vdots \\
I_R
\end{bmatrix},
\]

and

\[
W = \begin{bmatrix}
\gamma(1) a_{11}^S & \gamma(1) a_{12}^S & \cdots & \gamma(1) a_{1R}^S \\
\gamma(2) a_{21}^S & \gamma(2) a_{22}^S & \cdots & \gamma(2) a_{2R}^S \\
\vdots & \vdots & \ddots & \vdots \\
\gamma(S) a_{S1}^S & \gamma(S) a_{S2}^S & \cdots & \gamma(S) a_{SR}^S
\end{bmatrix},
\]

where each block is of size \( R \times R \), making \( W \) a (row stochastic) matrix of size \((RS) \times (RS)\), and \( w_{ij} = a_{ij}^S \) with \( i, j = 1, \ldots, RS \).

**Remark:** While Proposition 5 maps a given clMMMC to an HMM which from the outside looks the same as, this mapping is not reversible: not every HMM represents a clMMMC, and most importantly, parameter estimation algorithms such as the standard Baum-Welch algorithm cannot be used to estimate the parameters of a clMMMC, because they do not “respect the structure” of the matrix \( W \): the HMM \( \lambda \) is defined by \((RS)^2 - RS + (RS - 1)\) free parameters (the entries of \( W \) and the entries of \( \pi \) with the stochasticity constraints taken into account), whereas the corresponding clMMMC requires only \((pS - 1)(S - 1) + (SR - 1)(R - 1)\) parameters\(^9\).

Hence, it is not possible to estimate the parameters of \( \lambda \) and then compute the ones of \( \mu \); instead, we develop here a Baum-Welch-like algorithm to estimate the parameters of \( \mu \) directly.

5.1. Related literature on HMMs. The literature on estimating HMMs from the data is very rich. In what follows we refer to some of the papers which appear to be quite close to the subject of our work. A concept of regime switching time series models is widely used in econometrics \([10]\); these models allow parameters of the conditional mean and variance to vary according to some finite-valued stochastic process with states or regimes, similarly to the clMMMC proposed here. However, the observations are assumed to be generated by a deterministic process with random noise, and the switching process \( S \) is either a Markov chain independent of the past observations (like in the olMMMC) or is a deterministic function of the past observations. In contrast, a clMMMC is modulated by a Markov chain which depends on the past observations \((2)\). We stress that open loop MMMC are closely related to the so called Markov regime models for switching regressions \([13]\), where the Markov chain is used to select a density of an observed random variable, and EM type algorithm is then used to estimate the parameters of the densities from a given class, say Gaussian densities.

Another recent concept in which a latent Markov chain selects from a set of parametrisations of a visible process appears in e.g. \([17]\); in contrast to a clMMMC, the Markov chain is independent of the realisations of the output, and the visible process is static, subject to noise, not a Markov chain itself.

---

\(^8\)\( [\cdot] \) and \( j_R(\cdot) \) are defined in the proof, \((32)\).

\(^9\) Note that \((pS - 1)(S - 1) + (SR - 1)(R - 1) < (RS)^2 - 1 \) for \( R + S > 2 \).
**Identifiability:** Given the close relationship between cIMMMCs and HMMs outlined above, one should expect that identifiability issues for cIMMMCs bear close resemblance to those of HMMs. By identifiability we mean the following: assume that \((r_1 \ldots r_T)\) has been generated by the “true model” \(\mu^{\text{true}} \in \Lambda\); under which conditions and in what sense will the estimate \(\hat{\mu}^{\text{est}}\) defined by (15) converge to \(\mu^{\text{true}}\) if \(T \to \infty\)? For HMMs this question was partially answered in [20], namely it was shown that there is an open, full-measure subset \(U\) of all HMMs, such that the sequence of estimates of the BW algorithm converges to \(\lambda^{\text{true}}\) (or a trivial permutation of it), provided the starting model is chosen within \(U\), \(\lambda^{\text{true}}_i > \delta > 0\) and \(T \to \infty\). However, the structure of \(U\) and convergence speed in terms of the number of samples were not described, and, to the best of our knowledge, these questions are still open. For cIMMMCs, similarly and trivially, any permutation of \(A^S\) and the corresponding pages of \(A^R\), which amounts to relabelling the hidden states \(s\), yields the same visible process.

However, there are examples of sets of HMMs \(\lambda\), which are not permutations of each other, yet generate the same observable process; see [4,8]. Interestingly, those examples involve the special case of partially observable Markov chains, a subclass of HMMs with emissions matrices \(B\) having entries that are either 1 or 0. Comparing to Proposition 5, a cIMMMC has close correspondence to an HMM in general, we cannot recover the true model \(\mu^{\text{true}}\); our numerical experiments also suggest that, in particular, the specific contribution of this paper is to develop techniques in which unbiased user models are recovered in the case where recommenders, users, and switching between them can be parameterised in a Markovian manner, and where users and recommenders form part of a feedback system. Examples are given to present the efficacy of our approach.

### 6. Concluding remarks

We consider the identification of user models acting under the influence of one or more recommender systems. As we have already discussed, acting behavioural change affects the original model underpinning the predictor, leading to a biased user models. Given this background, the specific contribution of this paper is to develop techniques in which unbiased estimates of user behaviour can be recovered in the case where recommenders, users, and switching between them can be parameterised in a Markovian manner, and where users and recommenders form part of a feedback system. Examples are given to present the efficacy of our approach.

### Appendix A. Implementation Issues: Scaling

Since the computations of \(\alpha_t\) and \(\beta_t\) involve multiplications of on the order of \(2t\) numbers less than 1, for large \(T\), they will be close to, or below, machine precision. The re-estimation (15) then requires division of very small numbers, which of course should be avoided. A simple rescaling of \(\alpha_t\) and \(\beta_t\) mitigates these issues while requiring only minimal changes.

At every iteration, scale \(\alpha_t\) to sum up to 1:

\[
\alpha_t := 1^T \alpha_t \quad \quad \hat{\alpha}_t := \alpha_t / c_t.
\]

The update for \(\hat{\alpha}_t\) then becomes:

\[
\hat{\alpha}_t = \left( A^S(\gamma(r_{t-1})) \right)^T \left( a^R_{r_{t-1}r_t}(:) \circ (\alpha_{t-1} \hat{\alpha}_{t-1}) \right) / c_t \\
= \left( A^S(\gamma(r_{t-1})) \right)^T \left( a^R_{r_{t-1}r_t}(:) \circ \hat{\alpha}_{t-1} \right) c_{t-1} / c_t \\
= \left( A^S(\gamma(r_{t-1})) \right)^T \left( a^R_{r_{t-1}r_t}(:) \circ \hat{\alpha}_{t-1} \right) 1 / \left( A^S(\gamma(r_{t-1})) \right)^T \left( a^R_{r_{t-1}r_t}(:) \circ \alpha_{t-1} \right) / c_{t-1} \\
= \left( A^S(\gamma(r_{t-1})) \right)^T \left( a^R_{r_{t-1}r_t}(:) \circ \hat{\alpha}_{t-1} \right) 1 / \left( A^S(\gamma(r_{t-1})) \right)^T \left( a^R_{r_{t-1}r_t}(:) \circ \alpha_{t-1} \right).
\]
or in two steps:

\[
\hat{\alpha}_t' = (A^S(\gamma(r_{t-1})))^T (a^R_{r_{t-1}r_t}(i) \circ \hat{\alpha}_{t-1}) \quad \quad \quad \hat{\alpha}_t = \frac{\hat{\alpha}_t'}{\|\hat{\alpha}_t'\|}.
\]

Also note that

\[
\frac{c_{t-1}}{c_t} = \frac{1}{\|\hat{\alpha}_t'\|} \Rightarrow \log c_t = \log c_{t-1} + \log 1^T \hat{\alpha}_t'
\]

which is important to keep track of, because the log-likelihood of the model \( \mu \) is now given by

\[
\log \ell(\mu) = \log(1^T \alpha_T) = \log(c_T).
\]

Since the backwards variables \( \beta_{T-t} \) can be expected to be of similar order as \( \alpha_t \) they are scaled using the same scaling factors \( c_t \):

\[
\hat{\beta}_{T-t} := \beta_{T-t}/c_t
\]

Finally, we need to adapt the re-estimation formulas to the scaled variables. To this end, let

\[
\hat{\xi}(i, j) := \hat{\alpha}_{t-1}(i) a^R_{r_{t-1}r_t}(i) a^S_{i,j}(\gamma(r_{t-1})) \hat{\beta}_t(j)
\]

and note that then \( \xi_t(i, j) = c_{t-1}c_{T-t} \hat{\xi}(i, j) \); however, \( \sum_{i,j} \xi_t(i, j) = 1^T \alpha_T = c_{t-1}c_{T-t} \sum_{i,j} \hat{\xi}(i, j) \).

Hence, if we let

\[
\hat{\xi}_t := \frac{\hat{\xi}_t'}{1^T \hat{\xi}_t' 1} = \frac{\xi_t}{1^T \alpha_T}
\]

we can substitute \( \xi_t = (1^T \alpha_T) \hat{\xi}_t \) in (15) and note that \( 1^T \alpha_T \) cancels everywhere, so we finally arrive at the rescaled reestimation equations

\[
\begin{align*}
\pi_t^{S'} &= \sum_{j=1}^S \hat{\xi}_t(i, j) \\
 a_{ij}^{S'}(l) &= \frac{\sum_{\ell=\gamma(r_{t-1})=i} \hat{\xi}_t(i, j)}{\sum_{k=1}^S \sum_{\ell=\gamma(r_{t-1})=k} \xi_t(i, k)} \\
 a_{mn}^{R'}(i) &= \frac{\sum_{k=1}^S \sum_{l=r_{t-1}=m, r_t=n} \hat{\xi}_t(i, k)}{\sum_{k=1}^S \sum_{l=r_{t-1}=m} \hat{\xi}_t(i, k)}.
\end{align*}
\]

**Appendix B. Proofs.**

**Proof of Lemma 2.** Take \( \mu, \mu' \in \Lambda \). If \( \ell(\mu) = 0 \) then \( 10 \) \( Q(\mu, \mu') = Q(\mu, \mu) = 0 \), and so (14) holds. If \( \ell(\mu) > 0 \) then:

\[
\log \frac{\ell(\mu')}{\ell(\mu)} = \log \left( \sum_{s_0, \ldots, s_T} \frac{P(s_0, \ldots, s_T | \mu)}{\ell(\mu)} \frac{P(s_0, \ldots, s_T | \mu')}{P(s_0, \ldots, s_T | \mu)} \right) \\
\geq \sum_{s_0, \ldots, s_T} \frac{P(s_0, \ldots, s_T | \mu)}{\ell(\mu)} \log \frac{P(s_0, \ldots, s_T | \mu')}{P(s_0, \ldots, s_T | \mu)} = \frac{Q(\mu, \mu') - Q(\mu, \mu)}{\ell(\mu)}
\]

where we made use of \( \sum_{s_0, \ldots, s_T} \frac{P(s_0, \ldots, s_T | \mu)}{\ell(\mu)} = 1 \), and so in the second line we have a convex combination to which we can apply Jensen’s inequality. Note that the sums run over all \( s_0 \ldots s_T \) such that \( P(s_0, \ldots, s_T, r_0, \ldots, r_T | \mu) > 0 \) \( \Box \)

\(10\) Recall that \( 0(\pm \infty) = 0 \).
Proof of Lemma 1. Note that herein, summation indices $s, r$ always run from 1 to $S$. It follows from (3) that

$$
\log P(s_0, \ldots, r_T \mid \mu') = \log \pi_{r_0}^{R'} + \log \pi_{s_0}^{S'} + \sum_{t=1}^T \left( \log a_{s_{t-1}s_t}(\gamma(r_{t-1})) + \log a_{r_{t-1}r_t}(s_{t-1}) \right).
$$

Note that

$$
\xi_t(i, j) = P(S_{t-1} = i, S_t = j, r_0, \ldots, r_T \mid \mu) = \sum_{s_0, \ldots, s_1, s_{t-1}, s_t, \ldots, s_T} P(s_0, \ldots, s_1, s_{t-1} = i, S_t = j, s_2, \ldots, r_T \mid \mu),
$$

i.e. to get $\xi_t(i, j)$ we “marginalize out” all the states of the latent chain $S$ before time instant $t - 1$ and after time instant $t$. We compute:

$$
\sum_{s_0, \ldots, s_T} P(s_0, \ldots, r_T \mid \mu) \log \pi_{s_0}^{S'} = \sum_{i=1}^S \sum_{j=1}^S \log \pi_i^{S'} \sum_{s_2, \ldots, s_T} P(S_0 = i, S_1 = j, s_2, \ldots, r_T \mid \mu) = \sum_{i=1}^S \sum_{j=1}^S \xi_1(i, j)
$$

Define $L_{ijmn} := \log a_{ij}^{S'}(\gamma(m)) + \log a_{mn}^{R'}(i)$ and use the “marginalisation trick” again to compute

$$
\sum_{s_0, \ldots, s_T} P(s_0, \ldots, r_T \mid \mu) \sum_{t=1}^T L_{s_{t-1}s_tr_{t-1}r_t} = \sum_{t=1}^T \sum_{s_1, s_t} L_{s_{t-1}s_tr_{t-1}r_t} P(S_{t-1} = s_{t-1}, S_t = s_t, r_0, \ldots, r_T \mid \mu) = \sum_{t=1}^T \sum_{i=1}^S \sum_{j=1}^S L_{ijmnr_t} \xi_t(i, j).
$$

To complete the proof we note that $\sum_{s_0, \ldots, s_T} P(s_0, \ldots, r_T \mid \mu) \log \pi_{r_0}^{R'} = \ell(\mu) \log \pi_{r_0}^{R'}$. 

Proof of Theorem 3. From the remark before the theorem, it should be clear that we can ignore the first term in (11) in the maximisation. Consider $\mu \in \Lambda$ as fixed, and define $\tilde{\mu} \mapsto W(\tilde{\mu}) := Q(\mu, \tilde{\mu})$. We claim that $W$ has the unique global maximum point $\mu' \in \Lambda$. Note that if $\mu \in \Lambda$ then it may have zero components, say $\mu_{i_1} = \cdots = \mu_{i_d} = 0$. Then the logarithms of the corresponding components $\tilde{\mu}_{i_1} \ldots \tilde{\mu}_{i_d}$ of $\tilde{\mu}$ in $W$ are multiplied by 0 so that these components do not change $W$. However, if we fix all the components of $\tilde{\mu}$ but $\tilde{\mu}_{i_1}$ and any other component $\tilde{\mu}_k$ such that $k \not\in \{i_1 \ldots i_d\}$, and $k$ is such that $\tilde{\mu}_i$ and $\tilde{\mu}_k$ are in $\pi^R$, or are in the same row of $A^R$ or $A^S$, then increasing $\tilde{\mu}_{i_1}$ will decrease $\tilde{\mu}_k$ (to meet the stochasticity constraints). As a result, the log $\tilde{\mu}_k$ will decrease causing $W$ to decrease. Hence, the maximum of $W$ is attained in the set $\tilde{\Lambda} := \{\tilde{\mu} \in \Lambda : \tilde{\mu}_{i_1} = \cdots = \tilde{\mu}_{i_d} = 0\}$. Let $\tilde{W}$ be the restriction of $W$ to $\tilde{\Lambda}$. Now $\tilde{W}$ is a conical sum of logarithms of all $R + S + SR^2 + pS^2 - d$ independent components of $\tilde{\mu}$, hence strictly concave function on a convex compact set $\tilde{\Lambda}$. Hence, $\tilde{W}$ has a unique maximum point in $\tilde{\Lambda}$ which coincides with the unique global maximum point $\mu'$ of $W$ in $\Lambda$.

Let us prove that $\mu' = M(\mu)$ As noted above, $\mu_{i_1} = \cdots = \mu_{i_d} = 0$ implies that $\mu'_{i_1} = \cdots = \mu'_{i_d} = 0$, and we stress that the same property holds true for $M(\mu)$: If $\pi^S$ is 0 it follows from (6) that $a_{0}(i) = 0$ and from (13) we get $\xi_1(i, j) = 0$. By (15a), $\pi_i^S = 0$ as well. If we have $a_{ij}^{S}(t) = 0$, then again from (13), we see that $\xi_t(i, j) = 0$ for all $t$ with $r_{t-1} = l$, and (15b) yields $a_{ij}^{R}(l) = 0$. Similarly, $a_{mn}^{R}(i) = 0$ leads to $\xi_t(i, j) = 0$ whenever $r_{t-1} = m$ and $r_t = n$, independently of $j$, so $a_{mn}^{R}(i) = 0$, otherwise.
too. Hence \( M(\tilde{\Lambda}) = \tilde{\Lambda} \), and \( \pi^S_i \), \( a^S_{ij}(l) \) and \( a^R_{mn}(i) \) defined by \((15)\) are positive if the corresponding components of \( \mu \) are. On the other hand, \( \mu'_{k} > 0 \) if \( \mu_k > 0 \) as otherwise \( W(\mu') = -\infty \), and so the gradient of \( \tilde{W} \) is well-defined at \( \mu' \). In fact, for positive \( \tilde{\pi}^S_i \), \( \tilde{a}^S_{ij}(l) \) and \( \tilde{a}^R_{mn}(i) \):

\[
\frac{\partial \tilde{W}}{\partial \tilde{\pi}^S_i} = \frac{\sum_{j=1}^{S} \xi_1(i, j)}{\tilde{\pi}^S_i} \\
\frac{\partial \tilde{W}}{\partial \tilde{a}^S_{ij}(l)} = \frac{\sum_{t;\gamma(r_{t-1})=l} \xi_1(i, j)}{\tilde{a}^S_{ij}(l)} \\
\frac{\partial \tilde{W}}{\partial \tilde{a}^R_{mn}(i)} = \frac{\sum_{k=1}^{S} \sum_{t;\gamma(r_{t-1})=l} \xi_1(i, k)}{\tilde{a}^R_{mn}(i)}
\]

By e.g. [3, p.113, Prop. 2.1.2], it is necessary and sufficient for \( \mu' \) to satisfy the inequality

\[(28) \quad (\nabla \tilde{W}(\bar{\mu}) \bigg|_{\bar{\mu}=\mu'})^T \bar{\mu} \leq (\nabla \tilde{W}(\bar{\mu}) \bigg|_{\bar{\mu}=\mu'})^T \mu' \quad \forall \bar{\mu} \in \tilde{\Lambda};
\]

We stress that the r.h.s. of \((28)\) is independent of \( \mu' \) as

\[(29) \quad (\nabla \tilde{W}(\bar{\mu}) \bigg|_{\bar{\mu}=\mu'})^T \mu' = \sum_{i=1}^{S} \sum_{j=1}^{S} \xi_1(i, j) + \sum_{l=1}^{p} \sum_{i,k=1}^{S} \sum_{t;\gamma(r_{t-1})=l} \xi_1(i, j) + \sum_{l=1}^{S} \sum_{m=1}^{R} \sum_{t;\gamma(r_{t-1})=l} \xi_1(i, k) = (2T + 1) \ell(\mu)
\]

Since \( \pi^S_i \), \( a^S_{ij}(l) \) and \( a^R_{mn}(i) \) defined by \((15)\) are positive if the corresponding components of \( \mu \) are so, the gradient of \( \tilde{W} \) is well-defined at \( M(\mu) \). Take any \( \bar{\mu} \in \tilde{\Lambda} \) and compute:

\[
\tilde{\pi}^S_i \frac{\partial \tilde{W}}{\partial \tilde{\pi}^S_i}(\mu') = \tilde{\pi}^S_i \sum_{j=1}^{S} \xi_1(i, j) \\
\tilde{a}^S_{ij}(l) \frac{\partial \tilde{W}}{\partial \tilde{a}^S_{ij}(l)}(\mu') = \tilde{a}^S_{ij}(l) \sum_{k=1}^{S} \sum_{t;\gamma(r_{t-1})=l} \xi_1(i, k) \\
\tilde{a}^R_{mn}(i) \frac{\partial \tilde{W}}{\partial \tilde{a}^R_{mn}(i)}(\mu') = \tilde{a}^R_{mn}(i) \sum_{n=1}^{S} \sum_{k=1}^{R} \sum_{t;\gamma(r_{t-1})=l} \xi_1(i, k)
\]

so that, by stochasticity constraint, we get:

\[
(\nabla \tilde{W}(\bar{\mu}) \bigg|_{\bar{\mu}=\mu'})^T \bar{\mu} = \left( \sum_{i=1}^{S} \tilde{\pi}^S_i \right) \sum_{j=1}^{S} \xi_1(i, j) \\
+ \sum_{i=1}^{S} \sum_{l=1}^{p} \left( \sum_{j=1}^{S} \tilde{a}^S_{ij}(l) \right) \sum_{k=1}^{S} \sum_{t;\gamma(r_{t-1})=l} \xi_1(i, k) \\
+ \sum_{m=1}^{R} \sum_{n=1}^{S} \left( \sum_{i=1}^{R} \tilde{a}^R_{mn}(i) \right) \sum_{n=1}^{S} \sum_{k=1}^{R} \sum_{t;\gamma(r_{t-1})=l} \xi_1(i, k) = (2T + 1) \ell(\mu)
\]

Hence, \( M(\mu) \) defined by \((15)\) satisfies \((28)\) with equality for any \( \bar{\mu} \in \tilde{\Lambda} \). This completes the proof. \( \blacksquare \)
Proof of Corollary 4. Theorem 3 implies that \( Q(\mu, M(\mu)) \geq Q(\mu, \mu) > -\infty \) (as \( Q(\mu, \mu) > -\infty \) for any \( \mu \in \Lambda \)). Since \( Q(\mu, \cdot) \) has the unique maximum point \( M(\mu) \), it follows that \( Q(\mu, M(\mu)) > Q(\mu, \mu) = -\infty \) provided \( M(\mu) \neq \mu \). This implies that

\[
Q(\mu, M(\mu)) = Q(\mu, \mu) \iff M(\mu) = \mu.
\]

By (14), \( Q(\mu, M(\mu)) > Q(\mu, \mu) \) implies \( \ell(M(\mu)) > \ell(\mu) \). On the other hand, if \( \ell(M(\mu)) = \ell(\mu) > 0 \), then \( \log \frac{\ell(M(\mu))}{\ell(\mu)} = 0 \), and so, by (14), \( Q(\mu, M(\mu)) = Q(\mu, \mu) \). The latter and (30) implies that \( M(\mu) = \mu \). Hence

\[
\forall \mu \in \Lambda : \text{ either } \ell(M(\mu)) > \ell(\mu) \text{ and } M(\mu) \neq \mu \text{ or } \ell(M(\mu)) = \ell(\mu) \text{ and } M(\mu) = \mu.
\]

Now, by (31), if \( \mu_n \neq M(\mu_n) \) then \( \ell(M(\mu_n)) > \ell(\mu_n) \), and since \( \ell(\mu) \leq 1 \) for any \( \mu \in \Lambda \) it follows that \( \{\mu_n\} \) must have a limiting point \( \mu^\infty \) as otherwise we would get that \( \ell(\mu_k) > 1 \) for a finite \( k \). If \( M(\mu^\infty) \neq \mu^\infty \) then \( \ell(M(\mu^\infty)) > \ell(\mu^\infty) \), and we again arrive at the conclusion that \( \ell(M(\mu^\infty)) \neq \mu^\infty \).

Finally, to prove (16) let us first note that the gradient of \( Q \) at a fixed point \( \mu^\infty \) of \( M \) coincides with \( \nabla \ell(\mu^\infty) \). Indeed, by (10) it follows that:

\[
\partial_{\mu_i} Q(\mu, \mu') = \sum_{s_0, \ldots, s_T} P(s_0, \ldots, s_T, r_0, \ldots, r_T \mid \mu) \frac{\partial_{\mu_i} P(s_0, \ldots, s_T, r_0, \ldots, r_T \mid \mu')}{P(s_0, \ldots, s_T, r_0, \ldots, r_T \mid \mu')}
\]

Given \( \mu_0 \in \Lambda \) define \( \mu_k := M^{(k)}(\mu_0) \). At least one sub-sequence of \( \{\mu_k\} \), say \( \{\mu_n\} \) converges to \( \mu^\infty \): \( \lim_n \mu_n = \mu^\infty \). Note that

\[
\lim_n \partial_{\mu_i} Q(\mu^\infty, \mu') \bigg|_{\mu' = \mu_n} = \lim_n \sum_{s_0, \ldots, s_T} P(s_0, \ldots, s_T, r_0, \ldots, r_T \mid \mu^\infty) \frac{\partial_{\mu_i} P(s_0, \ldots, s_T, r_0, \ldots, r_T \mid \mu')}{P(s_0, \ldots, s_T, r_0, \ldots, r_T \mid \mu')} = \sum_{s_0, \ldots, s_T} \partial_{\mu_i} P(s_0, \ldots, s_T, r_0, \ldots, r_T \mid \mu') \bigg|_{\mu' = \mu^\infty} = \partial_{\mu_i} \ell(\mu) \bigg|_{\mu = \mu^\infty}
\]

Recalling the definition of \( \tilde{W} \) from the proof of the theorem, and noting that \( \nabla \tilde{W}(\mu) \bigg|_{\mu = \mu^\infty} = \nabla \ell(\mu) \bigg|_{\mu = \mu^\infty} \), we conclude that (16) follows from (28).  

Proof of Proposition 5. We need to show that \( P(r_0 \cdots r_T \mid \mu) = P(r_0 \cdots r_T \mid \lambda) \), where the notation means the probability of observing the sequence \( (R_0 = r_0, R_1 = r_1, \ldots, R_T = r_t) \) given that the sequence is generated by the cIMMC \( \mu \) or the HMM \( \lambda \), respectively.
We have

\[ P(r_0 \cdots r_T \mid \mu) = P(r_0)P(r_1 \mid r_0) \cdots P(r_T \mid r_{T-1}) \]

\[ = \prod_{s_0=1}^{S} \pi_{s_0}^{R} \sum_{s_0=1}^{S} \pi_{s_0}^{S}(r_1 \mid s_0, r_0) \prod_{s_2=1}^{S} P(s_2 \mid s_1, r_1)P(r_3 \mid s_2, r_2) \cdots \]

\[ \sum_{s_{T-1}=1}^{S} P(s_{T-2}, r_{T-2})P(r_T \mid s_{T-1}, r_{T-1}) \sum_{s_T=1}^{S} P(s_T \mid s_{T-1}, r_{T-1}) \]

\[ = \prod_{s_0=1}^{S} \pi_{s_0}^{R} \sum_{s_0=1}^{S} \pi_{s_0}^{S} \prod_{t=1}^{T} P(s_t \mid s_{t-1}, r_{t-1})P(r_t \mid s_{t-1}, r_{t-1}) \]

\[ = \prod_{s_0=1}^{S} \pi_{s_0}^{R} \prod_{t=1}^{T} a_{s_{t-1} s_t}^{S}(\gamma(r_{t-1}))a_{r_{t-1} r_t}^{R}(s_{t-1}) \]

\[ = \prod_{s_0=1}^{S} \pi_{s_0}^{R} \prod_{t=1}^{T} a_{s_{t-1} s_t}^{S}(\gamma(r'_{t-1}))a_{r'_{t-1} r'_t}^{R}(s_{t-1}) \delta_{r_t r'_t} \]

\[ = \prod_{q_0=1}^{T} W_{q_{t-1} q_t} b_{q_t r_t} = P(r_0 r_1 \cdots r_T \mid \lambda), \]

where

\[ \sum_{s_{T-1}=1}^{S} P(s_T \mid s_{T-1}, r_{T-1}) \]

\[ \text{equals to one and is added in the second expression for symmetry reasons;} \]

\[ \text{We use the Kronecker symbol } \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases} ; \]

\[ q_t \text{ denotes the (hidden) state of the HMM } \lambda, \text{ whereas its emissions are denoted by } r_t ; \]

\[ q \in [RS] \text{ and } (s, r) \in [S] \times [R] \text{ are related by the bijections} \]

\[ (s, r) \mapsto (s-1)R + r \]

\[ q \mapsto ([q/R], j_R(q)), \]

with

\[ j_R(k) := (k-1)(\text{mod } R) + 1 \]

\[ [p] := \inf \{ k \in \mathbb{Z} \mid k \geq p \}. \]

**Appendix C. Multiple (short) Trajectories.** Let us assume that we observe \( D \) full trajectories \( \mathcal{O}^1, \ldots, \mathcal{O}^D, \mathcal{O}^d = (r_0^d, r_1^d, \ldots, r_{T_d}^d) \), each having a potentially different length \( T_d \). As above, define the individual likelihoods and auxiliary functions for each \( \mathcal{O}^d \) by

\[ \ell^d(\mu) := P(\mathcal{O}^d \mid \mu) = P(R_0 = t_0^d, \ldots, R_{T_d} = t_{T_d}^d \mid \mu) \]

\[ Q^d(\mu, \mu') := \sum_{s_0 \ldots s_{T_d}} P(\mathcal{O}^d, s_0, \ldots, s_{T_d} \mid \mu) \log P(\mathcal{O}^d, s_0, \ldots, s_{T_d} \mid \mu'). \]

The following lemma provides representation for the likelihood and auxiliary function \( Q \) for the case of multiple observations.
Lemma 6.

\[(34) \quad \ell(\mu) = P(O^1, \ldots, O^D | \mu) = \prod_d \ell^d(\mu)\]

\[(35) \quad Q(\mu, \mu') := \sum_{s_0^1, \ldots, s_{T_d}^D} P(O^1, \ldots, O^D, s_0^1, \ldots, s_{T_d}^D | \mu) \log P(O^1, \ldots, O^D, s_0^1, \ldots, s_{T_d}^D | \mu')\]

\[(36) \quad = \sum_d \frac{\ell(\mu)}{\ell^d(\mu)} Q^d(\mu, \mu').\]

**Proof.** While (34) follows easily from the assumption that the trajectories are mutually independent, to prove that (35) equals (36) we require a few additional algebraic manipulations. First, let us note that since the observed trajectories $O^1, \ldots, O^D$ are mutually independent, it follows that: $P(O^1, \ldots, O^D, s_0^1, \ldots, s_{T_d}^D | \mu) = \prod_d P(O^d, s_0^d, \ldots, s_{T_d}^d | \mu)$. Introduce for convenience the abbreviation $J_d(\mu) := P(O^d, s_0^d, \ldots, s_{T_d}^d | \mu)$, and rewrite (35) as

\[
Q(\mu, \tilde{\mu}) = \sum_{s_0^1, \ldots, s_{T_d}^D} \prod_d \frac{P(O^d, s_0^d, \ldots, s_{T_d}^d | \mu)}{J_d(\mu)} \left[ \sum_{d' = 1}^D \log P(O^{d'}, s_0^{d'}, \ldots, s_{T_d}^{d'} | \mu) \right]
\]

\[
= \sum_{s_0^1, \ldots, s_{T_d}^D} \left\{ \prod_d J_d(\mu) \left[ \sum_{d'} \log J_{d'}(\tilde{\mu}) \right] \right\}
\]

\[
= \sum_{s_0^1, \ldots, s_{T_d}^D} \left\{ \log J_1(\tilde{\mu}) \left( \prod_d J_d(\mu) \right) + \log J_2(\tilde{\mu}) \left( \prod_d J_d(\mu) \right) + \cdots \right\}
\]

\[
= \sum_{s_0^1, \ldots, s_{T_d}^D} \left\{ \log J_1(\tilde{\mu}) J_1(\mu) \left( \prod_d J_d(\mu) \right) + \log J_2(\tilde{\mu}) J_2(\mu) \left( \prod_{d \neq 2} J_d(\mu) \right) + \cdots \right\}
\]

\[
= \sum_{s_0^1, \ldots, s_{T_d}^D} \sum_{s_0^1, \ldots, s_{T_d}^D} \left\{ \log J_1(\tilde{\mu}) J_1(\mu) \left( \prod_d J_d(\mu) \right) + \sum_{d \neq 2} \log J_d(\mu) \left( \prod_{d \neq 2} J_d(\mu) \right) + \cdots \right\}
\]

\[
= Q^1(\mu, \tilde{\mu}) \prod_{d \neq 1} \left( \sum_{s_0^d, \ldots, s_{T_d}^d} J_d(\mu) \right) + Q^2(\mu, \tilde{\mu}) \prod_{d \neq 2} \left( \sum_{s_0^d, \ldots, s_{T_d}^d} J_d(\mu) \right) + \cdots
\]

\[
= Q^1(\mu, \tilde{\mu}) \prod_{d \neq 1} \ell^d(\mu) + Q^2(\mu, \tilde{\mu}) \prod_{d \neq 2} \ell^d(\mu) + \cdots
\]

\[
= \sum_d \frac{\ell(\mu)}{\ell^d(\mu)} Q^d(\mu, \tilde{\mu}).
\]
To see that one indeed may proceed to (⋆), i.e. the product and the sum can be exchanged, it is sufficient to check that \(\prod_{d=1}^D \sum_{r=1}^{R_d} e^{d} = \sum_{r=1}^{R_1} \cdots \sum_{r=1}^{R_D} \prod_{d=1}^D e^{d}_r.\) The proof of the latter equality is sketched below:\(^{11}\)

\[
\prod_{d=1}^D \sum_{r=1}^{R_d} c^{d}_r = (c^1_1 + c^1_2 + \cdots + c^1_{R_1}) \cdot (c^2_1 + c^2_2 + \cdots + c^2_{R_2}) \cdots (c^D_1 + c^D_2 + \cdots + c^D_{R_D}) = \\
\sum_{r=1}^{R_1} c^{d}_r \left( \prod_{d=2}^D \sum_{r=1}^{R_d} c^{d}_r \right) = \cdots = \sum_{r=1}^{R_1} \cdots \sum_{r=1}^{R_D} c^{d}_r = \sum_{r=1}^{R_1} \cdots \sum_{r=1}^{R_D} \left( \prod_{d=1}^D c^{d}_r \right).
\]

Finally, by the law of total probability, we get:

\[
\sum_{s^d_0, \ldots, s^d_{T_d}} J_d(\mu) = \sum_{s^d_0, \ldots, s^d_{T_d}} P(O^d, s^d_0, \ldots, s^d_{T_d} | \mu) = P(O^d | \mu) = \ell^d(\mu).
\]

We now check that Lemma 2 still holds with the \(Q\) defined in (36):

\[
\log \frac{\ell(\mu)}{\ell(\tilde{\mu})} = \log \left( \frac{\sum \frac{P(O^1, \ldots, O^D, s^0_1, \ldots, s^D_{T_D} | \tilde{\mu})}{P(O^1, \ldots, O^D, s^0_1, \ldots, s^D_{T_D} | \mu)} P(O^1, \ldots, O^D, s^0_1, \ldots, s^D_{T_D} | \mu)}{\ell(\mu)} \right) \\
\geq \sum_{s^0_0, \ldots, s^D_{T_D}} \frac{P(O^1, \ldots, O^D, s^0_1, \ldots, s^D_{T_D} | \mu)}{\ell(\mu)} \left( \log P(O^1, \ldots, s^0_1 | \tilde{\mu}) - \log P(O^1, \ldots, s^0_1 | \mu) \right) \\
= \frac{1}{\ell(\mu)} \left( Q(\mu, \tilde{\mu}) - Q(\mu, \mu) \right).
\]

Hence, to maximize the likelihood (34) we can utilize the same approach as for the case of observations in the form of one long trajectory, namely to maximize the lower bound of the likelihood improvement, given by the function \(Q\).

C.1. Re-estimation. Utilizing Lemma 6 we derive the following result:

**Theorem 7 (Re-estimation for multiple trajectories).** The unique maximiser \(\mu' = M(\mu)\) of \(Q(\mu, \mu')\) defined in (36) is given by

\[
\pi^R_m = \frac{\text{card}\{d \mid r^d_0 = m\}}{D}, \\
\pi^s_i = \frac{\sum_d 1/\ell^d(\mu) \sum_j \xi^d(i, j)}{D}, \\
a^R_{mn}(i) = \frac{\sum_d 1/\ell^d(\mu) \sum_{r^d_1 = m, r^d_n = n} \sum_j \xi^d(i, j)}{\sum_d 1/\ell^d(\mu) \sum_{r^d_1 = m} \sum_j \xi^d(i, j)}, \\
a^s_{ij}(l) = \frac{\sum_d 1/\ell^d(\mu) \sum_{r^d_l' = 1} \sum_k \xi^d(i, j)}{\sum_d 1/\ell^d(\mu) \sum_{r^d_l' = 1} \sum_k \xi^d(i, j)}.
\]

where \(i, j, k = 1, \ldots, S, l = 1, \ldots, p, m, n = 1, \ldots, R, d = 1, \ldots, D\) and \(\xi^d(i, j)\) is as defined in (12) with \(r^d_t\) replaced by \(r^d_t\). Note that \(\pi^R_m\) does not depend on \(\mu\), hence there is no need to make its estimation part of the iteration – it is estimated only once.

\(^{11}\)One can check this as well by applying the standard induction argument.
Proof. We introduce the Lagrangian $L(\mu', \lambda)$ by

$$L(\mu', \lambda) := Q(\mu, \mu') + \lambda^T(\text{1}^T \pi^R - 1) + \lambda^T_2(\text{1}^T \pi^S - 1) + \sum_{i=1}^p \lambda^T_1(A_S(l)\text{1} - 1) + \sum_{m=1}^S \lambda^T_m(A_R(m)\text{1} - 1),$$

where, for convenience, we omitted (and will omit for the remainder of the proof) the prime on all parameters $A^R, A^S, \pi^R, \pi^S$. We compute

$$\frac{\partial L}{\partial \pi^R_m} = \sum_d \ell(\mu)/\ell(\mu) \left( \delta_{mr_d} \frac{\ell^d(\mu)}{\pi^R_m} \right) + \lambda^R_1 = \sum_{d: r_d^0 = m} \ell(\mu) + \lambda^R_1,$$

where we used (36) and Lemma 1 (with the superscript $d$ added on every $\ell, \xi, r, s$); $\delta_{ij}$ denotes the Kronecker symbol and encodes the fact that if $r_d^0 \neq m$, then $Q^d$ does not depend on $\pi^R_m$ and hence $\frac{\partial Q^d}{\partial \pi^R_m} = 0$. Multiplying each equation by $\pi^R_m$ and summing over all $m = 1, \ldots, R$ yields

$$\frac{\partial L}{\partial \pi^R_m} = 0 \Rightarrow \sum_m \sum_{d: r_d^0 = m} \ell(\mu) + \sum_m \pi^R_m \lambda^R_1 = 0 \iff \lambda^R_1 = -\ell(\mu) D,$$

where we used the stochasticity constraint $\sum_m \pi^R_m = 1$. Inserting this into (38) set to zero, we get

$$\pi^R_m = -\sum_{d: r_d^0 = m} \ell(\mu) \lambda^R_1 = \sum_{d: r_d^0 = m} \frac{1}{D} = \frac{\text{card}\{d \mid r_d^0 = m\}}{D}.$$

Following a similar process for $\pi^S_i$, we get

$$\frac{\partial L}{\partial \pi^S_i} = \sum_d \ell(\mu)/\ell(\mu) \sum_j \xi^d(i, j) \frac{\pi^S_i}{\pi^S_i} + \lambda^S_2$$

and

$$\sum_i \pi^S_i \left( \sum_d \ell(\mu)/\ell(\mu) \sum_j \xi^d(i, j) \frac{\pi^S_i}{\pi^S_i} + \lambda^S_2 \right) = \lambda^S_2 + \sum_d \ell(\mu)/\ell(\mu) \sum_{i,j} \xi^d(i, j) = \lambda^S_2 + D\ell(\mu),$$

where we used that $\sum_{i,j} \xi^d(i, j) = \sum_{i,j} P(S_{t-1} = i, S_t = j, r_d^{t-1}, r_d^T | \mu) = P(r_d^{t-1}, r_d^T | \mu) = \ell^d(\mu)$. Setting $\frac{\partial L}{\partial \pi^S_i} = 0$ then yields (37b). Analogously, we get

$$\frac{\partial L}{\partial a^R_m(i)} = \sum_d \ell(\mu)/\ell(\mu) \sum_n \sum_{j: r_d^{t-1} = m, r_d^T = n} \xi^d(i, j) / a^R_m(i) + [\lambda^R_1]_m$$

and

$$\sum_i [\lambda^R_1]_m = -\sum_d \ell(\mu)/\ell(\mu) \sum_n \sum_{j: r_d^{t-1} = m, r_d^T = n} \xi^d(i, j) = -\sum_d \ell(\mu)/\ell(\mu) \sum_j \sum_{i: r_d^{t-1} = m} \xi^d(i, j)$$

which, when combined, yield the rest of (37).
Again, we face the issue of scaling: In computing $\xi^d_t$, we need to compute $\alpha^d_t$ and $\beta^d_t$, which are products of smaller and smaller numbers. This is easily remedied by computing the scaled counterparts, $\hat{\xi}^d_t$, $\hat{\alpha}^d_t$, and $\hat{\beta}^d_t$. One might now expect an issue arising from $\ell^d(\mu)$, the likelihood, not being available anymore, but only its logarithm $\log(\ell^d(\mu))$, see (20). However, this issue goes away, because we have by (23) that

$$\hat{\xi}^d_t = \frac{\xi^d_t}{1^T \alpha^d_T} = \frac{\xi^d_t}{\ell^d(\mu)}$$

and hence we can restate (37b)-(37d) in scaled variables:

\begin{align*}
(37b) & \quad \pi^S_i = \frac{\sum_j \hat{\xi}^d_t(i,j)}{\sum_e D_{j,e}(i)} \\
(37c) & \quad \alpha^R_{mn}(i) = \frac{\sum_d \sum_{t: r^d_{m-1} = m, r^d_t = n} \sum_j \hat{\xi}^d_t(i,j)}{\sum_d \sum_{t: r^d_{m-1} = m} \sum_j \hat{\xi}^d_t(i,j)} \\
(37d) & \quad \alpha^S_{lj}(l) = \frac{\sum_d \sum_{t: \gamma(r^d_{l-1}) = l} \hat{\xi}^d_t(i,j)}{\sum_d \sum_{t: \gamma(r^d_{l-1}) = l} \sum_k \hat{\xi}^d_t(i,k)};
\end{align*}

this form avoids division or subtraction of huge or tiny numbers.

Remark: Above re-estimation formulae (37) and (37b)-(37d) simplify to the single-trajectory case (15) and (24) for $D = 1$. Equation (37a) for $\pi^R$ reduces to $\pi^R = \epsilon_{r_0}$, see also the remark just before Theorem 3.

C.2. Revisiting the example of Sec. 4.2. Now that we have at our disposal an algorithm that explicitly works with multiple short trajectories, we can revisit the example identifying a driver model from Section 4.2. There, we concatenated all the trajectories, introducing dummy arcs from the final nodes to the initial nodes, in order to apply the vanilla version of the EM algorithm to the problem of identifying the driver’s preferences. The additional information that full trips are planned either by the driver or the recommender, was introduced by leveraging the closed-loop framework and allowing the state of $S$, i.e. whether driver or recommender are active, to switch only when a full trip is finished.

Using instead (37), there is no need to concatenate, because we can now work with many trips instead of a single long one. However, a new subtle complication arises: because there are now terminal nodes (nodes 8 and 9), they are never transitioned from and hence would break the algorithm. We could either modify the algorithm to work with rectangular matrices – the rows corresponding to such terminal nodes would have to be removed – or we reintroduce the dummy arcs with probability one to return to the initial node. For now, we chose the second option, hence, a trip of $\mathcal{O}^d = (r^d_1, \ldots, r^d_6) = (1, 7, 4, 5, 8)$ will be augmented to $\mathcal{O}^d = (r^d_1, \ldots, r^d_5, r^d_6) = (1, 7, 4, 5, 8, 1)$. The initial estimates of rows 8 and 9 are set to $[1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0]$, and so there will be no additional re-estimation necessary. Note that it would also be possible to add self-loops to nodes 8 and 9 instead of arcs back to the origins; however, the resulting matrices $A^R(\ell)$ are then reducible, and have no uniquely defined stationary distribution anymore.

Another subtlety concerns $A^S$: leaving it as a parameter to be estimated corresponds to the open-loop case in Section 4.2, i.e. it is allowed for the decision maker to change from driver to recommender mid-trip. As can be seen in Figure 5, that leads to unsatisfactory results. Incorporating again the information that the decision maker does in fact not change mid-trip corresponds to choosing $A^S = I$, and the probability of recommender vs driver then is estimated by $\pi^S$.

In Figure 5, we show some results. We generated $N_e = 50$ datasets by generating $N_t = 80$ trips for each, just as described in Section 4.2. The results are very similar, and the estimate of $p_r$ is
Figure 5. Results from estimating the driver preferences in Section C.2. Assuming open loop again yields acceptable results but enforcing that only full trips are planned by either driver or recommender improves accuracy considerably again. Mean and standard deviation are given.

0.37(±0.10) for the open-loop case, and 0.30(±0.07) for the closed-loop case.

Acknowledgements. The authors would like to thank Ming-Ming Liu and Yingqi Gu (University College Dublin) for help with the numerical experiments, and Giovanni Russo and Jakub Mareček (IBM Research Ireland) for valuable discussions.

REFERENCES

[1] J. Angrist and J. Pischke, Mostly harmless econometrics: An empiricist’s companion, Princeton University Press, 2008.
[2] Y. Baum, T. Petri, G. Soules, and N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, The Annals of Mathematical Statistics, 41 (1970), pp. 164–171.
[3] D. P. Bertsekas, Nonlinear Programming, Athena Scientific Belmont, 1999.
[4] D. Blackwell and L. Koopmans, On the identifiability problem for functions of finite Markov chains, The Annals of Mathematical Statistics, (1957), pp. 1011–1015.
[5] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl, Is seeing believing?: How recommender system interfaces affect users’ opinions, in Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, 2003, pp. 585–592.
[6] E. Crisostomi, R. Shorten, and F. Wirth, Smart cities: A golden age for control theory? [industry perspective], IEEE Technology and Society Magazine, 35 (2016), pp. 23–24.
[7] Y. Ephraim and W. J. J. Roberts, An EM algorithm for Markov modulated Markov processes, IEEE Transactions on Signal Processing, 57 (2009), pp. 463–470, https://doi.org/10.1109/TSP.2008.2007919.
[8] E. Gilbert, On the identifiability problem for functions of finite Markov chains, Ann. Math. Stat., 30 (1959), pp. 688–697.
[9] J. Krumm, A Markov model for driver turn prediction, tech. report, SAE Technical Paper, 2008.
[10] T. Lange and A. Rahbek, An introduction to regime switching time series models, in Handbook of Financial Time Series, Springer, 2009.
[11] Y. Lassoued, J. Monteil, Y. Gu, G. Russo, R. Shorten, and M. Mevissen, Hidden Markov model for route
and destination prediction, in IEEE International Conference on Intelligent Transportation Systems, 2017.

[12] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*, American Mathematical Soc., 2 ed., 2009.

[13] G. Lindgren, *Markov regime models for mixed distributions and switching regressions*, Scand. J. Stat., 5 (1978), pp. 81–91.

[14] L. Ljung, *System Identification: Theory for the User*, Pearson Education, 1998, https://books.google.ie/books?id=fYSrk4wDKPsC.

[15] I. D. Loram, H. Gollee, M. Lakie, and P. J. Gawthrop, *Human control of an inverted pendulum: Is continuous control necessary? Is intermittent control effective? Is intermittent control physiological?*, The Journal of Physiology, 589 (2011), pp. 307–324, https://doi.org/10.1113/jphysiol.2010.194712, http://dx.doi.org/10.1113/jphysiol.2010.194712.

[16] Z. Lu and Q. Yang, *Partially observable Markov decision process for recommender systems*. arXiv:1608.07793v2.

[17] M. Meila and M. I. Jordan, *Markov mixtures of experts*, in Multiple Model Approaches to Nonlinear Modelling and Control, R. Murray-Smith and T. A. Johanssen, eds., Taylor and Francis, 1996.

[18] J. Norton, *An Introduction to Identification*, Dover Books on Electrical Engineering Series, Dover Publications, 2009, https://books.google.ie/books?id=eyHC7751ncC.

[19] J. Pearl, *Causality: Models, Reasoning, and Inference*, New York: Cambridge University Press, 2000.

[20] T. Petrie, *Probabilistic functions of finite state Markov chains*, Ann. Math. Stat., 40 (1969), pp. 97–115.

[21] L. R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*, Proceedings of the IEEE, 77 (1989), pp. 257–286, https://doi.org/10.1109/5.18626.

[22] R. Simmons, B. Browning, Y. Zhang, and V. Sadekar, *Learning to predict driver route and destination intent*, in 2006 IEEE Intelligent Transportation Systems Conference, Sept 2006, pp. 127–132, https://doi.org/10.1109/ITSC.2006.1706730.

[23] A. Sinha, D. Gleich, and K. Ramani, *Deconvolving feedback loops in recommender systems*, in Proceedings of NIPS, Barcelon, Spain, 2016. arXiv:1703.01049.

[24] T. Soderstrom and P. Stoica, *System Identification*, Prentice Hall International Series In Systems And Control Engineering, Prentice Hall, 1989, https://books.google.ie/books?id=X_xQAAAAMAAJ.

[25] F. Stock, James H.; Trebbi, *Retrospectives: Who invented instrumental variable regression?*, Journal of Economic Perspectives, 17 (2003), pp. 177–194.

[26] J. F. T. Hastie, R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer, 2001.

[27] H. R. Varian, *Causal inference in economics and marketing*, Proc. of the National Academy of Sciences of the United States of America, (2016), p. 73107315.