Learning Hidden Markov Models When the Locations of Missing Observations are Unknown

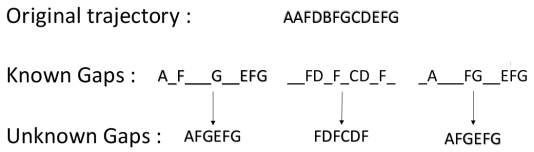
Binyamin Perets [*1] **Mark Kozdoba** [*1] **Shie Mannor** [1]

Abstract

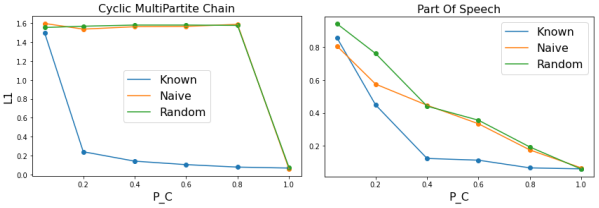
The Hidden Markov Model (HMM) is one of the most widely used statistical models for sequential data analysis. One of the key reasons for this versatility is the ability of HMM to deal with missing data. However, standard HMM learning algorithms rely crucially on the assumption that the positions of the missing observations *within the observation sequence* are known. In the natural sciences, where this assumption is often violated, special variants of HMM, commonly known as Silent-state HMMs (SHMMs), are used. Despite their widespread use, these algorithms strongly rely on specific structural assumptions of the underlying chain, such as acyclicity, thus limiting the applicability of these methods. Moreover, even in the acyclic case, it has been shown that these methods can lead to poor reconstruction. In this paper we consider the general problem of learning an HMM from data with unknown missing observation locations. We provide reconstruction algorithms that do not require any assumptions about the structure of the underlying chain, and can also be used with limited prior knowledge, unlike SHMM. We evaluate and compare the algorithms in a variety of scenarios, measuring their reconstruction precision, and robustness under model miss-specification. Notably, we show that under proper specifications one can reconstruct the process dynamics as well as if the missing observations positions were known.

1. Introduction

Hidden Markov Models (HMMs),(Murphy, 2012), are a well established and widely used method for modeling sequential data, with applications in a variety of fields, such as speech recognition (Yu & Deng, 2016) and economical time series (Hamilton, 2020; Kaufmann, 2019), to name a few. In all of the above applications, it is common to have *missing observations*. That is, we assume that a certain system, after N time steps, produces a full sequence of observations $U_i = (u_1, \ldots, u_N)$, but there is a *known* subset of times, $1 \leq t_1, \ldots, t_m \leq N$, $m \leq N$, such that the observation values at these times, $\{u_{t_j}\}_{j=1}^{m}$, are not known to us. This situation is illustrated in the first and second lines of Figure 1a. We refer to this type of missing values as missing values with *known locations* (or *known gaps*, Interchangeably), since we know which of the values are missing.



(a) Omission Process Illustration



(b) Ignoring the Gaps

Figure 1. **(a)** The gaps location may alter the sequence considerably. **(b)** The *Naive* approach may perform poorly, highlighting the need for a more careful treatment of gaps. (s.d $<$ .062)

1.1. Motivation and Previous Works

It is commonly assumed that observations are provided with time stamps, and that a single time unit corresponds to a step of the underlying chain. Such assumptions naturally imply that when there are missing values their location is known, however, this assumption do not hold for many applications. For example consider the case of irregular time series (Ramati.M, 2010), also known as unevenly spaced

[*]Equal contribution [1]Technion Israel Institute of Technology, Haifa, Israel. Correspondence to: Binyamin Perets <sbp67250@campus.technion.ac.il >.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

time series, where the system is observed at varying time intervals that do not correspond to the steps of the underlying chain. For example, in the context of cyber-attack detection, it is often assumed that the attacker sequence of actions can be modeled as a Markov process (Chadza et al., 2020). However, in practice, the observations of these malicious activities are irregular and incomplete. Another scenario is when the sampling clock is different from the underlying process clock. An example of this is the monitoring of sepsis patients, as the disease can progress in an unpredictable manner, resulting in inconsistent and possibly incomplete measurements of vital signs (Moor et al., 2019).

One domain where missing observations in unknown locations is frequently encountered is computational biology, where data is often only ordered and lacks time stamps(Chen et al., 2018; Liu et al., 2017b; Orr et al., 2018). For that reason, Silent-state HMM were developed. In SHMM, some states are not associated with any observation, and the model can transition between these states without emitting any observation. These states are referred to as "silent" states. Notice that silent states are incorporated explicitly into the assumed chain structure. For example, consider the wide use of Profile hidden Markov models (Eddy, 1998) for DNA sequence searching, where the goal is to find optimal matches between a model input and a database of sequences (Finn et al., 2011; Wheeler & Eddy, 2013). In this context, PHMMs are used to match the input model with sequences in the database that are assumed to come from the same model. However, it is also assumed that some states (nucleotides) may have been "deleted" through the course of evolution and are therefore not observable, or "silent".

Despite their potential usefulness and popularity, the uses of PHMMs and SHMMs are limited to a small set of problems. Currently, the focus of PHMM use is on inference, where it is assumed that the transition matrix and omitting probabilities are known (known as the profile HMM with known profiles). However, there is a significant need for many applications of SHMM to be able to directly learn the model parameters from data(Wheeler & Eddy, 2013; Setty et al., 2019; Ye et al., 2019; Orr et al., 2018; Pattabiraman & Warnow, 2021). There are currently several limitations that have hindered this goal. **First**, while the use of Expectation Maximization algorithms for learning the parameters of SHMM has been proposed, it has not yet been thoroughly studied and has not been shown to be effective in practice. In addition, it has been suggested in recent years that PHMMs are not identifiable (Pattabiraman & Warnow, 2021). In this paper, we address this limitation and examine the reasons for the failure of these algorithms, providing a new perspective on the identifiability issue. **Second**, while partial knowledge that is crucial for learning is often available, current frameworks are not flexible enough to incorporate this information. **Most importantly**, SHMMs assume that the underlying chain is acyclic (i.e the underline chain forms a DAG), which is frequently violated in the field of computational biology (Ye et al., 2019; Deconinck et al., 2021; Lummertz da Rocha et al., 2018). This assumption can be particularly problematic in areas such as pseudo-time analysis (Ye et al., 2019) in a variety of biological systems, (Campbell & Yau, 2018), with particularly extensive applications in single-cell trajectory analysis which is crucial in understanding the dynamics of a cell (see for instance (Chen et al., 2018; Van den Berge et al., 2020) and a survey (Deconinck et al., 2021)). In this type of applications, one is given a *set* of observations, usually corresponding to information about the state of a cell. However, due to the nature of the data collection process, it is often impossible to obtain timestamps for each individual observation.

In the interest of providing a comprehensive overview, we would like to highlight the following previous works related to this topic: In (Ramati.M, 2010), the authors formally described the concept of irregular time series in HMMs, but did not propose an algorithm for reconstruction. (Morimura et al., 2013) dealt with the sub-problem of flow network analysis where the positions of a few observed nodes are fixed and known in advance. The effort closest to our own, as far as we are aware, is the work of (Orr et al., 2018), in which the authors developed an Expectation Maximization-based algorithm for reconstructing HMMs from data with similar properties, but under the restriction of acyclic chains and a strict assumption about the underlying Markov process.

1.2. Missing Observations In Unknown Locations and The Need For Special Methods

Let $O = (o_1, \ldots, o_k)$ be an *order* set of observations, that is, we assume that a system passed through some sequence of states $S = (s_1, \ldots, s_N)$, *some* of which have generated the observations O. This situation is illustrated in the third line of Figure 1a, where different subsequences correspond to different possible values of O. Note in particular that the same O can result from different deletions of the full observation sequence. In this situation, we do not apriori know the positioning of the observations O (and, equivalently, of the observations missing from O) with respect to the state sequence S, and therefore we refer to such observations as observations with *unknown gaps*. The goal of the reconstruction (or learning) algorithm is to recover the underlying dynamics of the state sequence S from the observations O.

It is natural to ask whether one indeed needs to treat the unknown gaps situation specially. For example, a possible approach to reconstruction, termed *"Naive"* in this paper, is to simply ignore the missing observations and to reconstruct from sequences O as if these were the full observation sequences U. See for instance (Liu et al., 2017b; Setty et al., 2019). We now show that the *Naive* method can indeed

introduce extremely large errors into estimation. First, we denote $\Psi(s)$ to be the probability of observation emitting from state s to be omitted (see Section 2). For a fixed benchmark HMM and a fixed $\Psi(s) = 1 - p_c : \forall s$, we generate synthetic observation trajectories with gaps based on p_c, and compare three reconstruction methods: (a) The *Naive* approach, where we ignore the missing observations, (b) The Random approach, where we use the standard reconstruction algorithm with *known* gaps, but the gap locations given to the algorithm are chosen at random, and (c) The "Known" reconstruction, where the gap locations provided to the algorithm are the *true* gap locations. This algorithm is used as an **ideal benchmark**, as its performance is the best one can hope for, for any unknown gaps algorithm. The experiment is repeated for different benchmark HMM (see Section 4 for full details), and for a range of values of p_c. For each run, we measure the quality of the reconstruction by the L_1 distance between the reconstructed and the ground truth transition matrices.

As shown in Figure 1b, for intermediate values such as $p_c = 0.5$ the performance of the *Naive* and random approaches are poor, and for some (MultiPartite HMM) this is especially pronounced, with error of 1.6 even at $p_c = 0.8$ (the maximal error of L_1 is 2). This experiment emphasizes the need for reconstruction algorithms that model the unknown gap locations much more carefully than the *Naive* and "random" methods. Appendix A contains additional benchmark HMMs.

1.3. Paper Summary and Contributions

This paper aims to propose a comprehensive solution for learning HMMs in scenarios where observations are missing in unknown locations. Those solutions do not impose restrictive assumptions on the underlying chain, with the exception of the "Naive" method, which is shown to be ineffective. We demonstrate the effectiveness of our approach through various methods, which correspond to varying levels of prior knowledge about the process. These methods enable reconstruction in cases where prior knowledge is partially known, which is not possible with previous methods. The rest of the paper is organized as follows:

Model definition. Section 2 defines the HMMOP (HMM with Omission Process) model. In this model, a set U of full sequences of observations $U_i = (u_{i1}, \ldots, u_{iN})$ is generated by a standard HMM. Then, the sequence O_i is formed by deleting (equivalently, omitting) entries $u_{ij} \in U_i$ at random with a probability Ψ which depends on the state, independently of the other $u_{i'j'}$. This type of state dependent omitting process in known as **Non- Ignorable Omitting Process**. Section 2.1 examines the non-ignorable setup for missing observations, while Section 4.2 emphasizes the significance of taking $\Psi(\cdot)$ into account for accurate reconstruction.

Analytical Analysis. Section 2.2 presents an analytical analysis for the case when the probability of missing observations is constant across all states, $\Psi(s) = 1 - p_c \forall s$. This analysis highlights the limitations of previous attempts to reconstruct HMMs with missing values using PHMMs.

Reconstruction methods. Section 3 introduces two approaches to reconstructing HMMOPs. One approach assumes that for the ignorable setup, only the observed trajectories $\mathcal{O} = \{O_i\}$ and the length of the original sequence are known. The second approach assumes that only the omitting probabilities (Ψ) are known. Additionally, we present a method for imputing Ψ when only partial knowledge of it is available.

Experiments and evaluations. Section 4 evaluates the unknown gaps reconstruction algorithms on a number of synthetic and semi-synthetic benchmarks, where it is possible to compare the reconstructed transition matrices to the ground truth matrices that generated the data. The experiments are performed for a range of Ψ, allowing us to study the effect of different percentages of missing observations on the performance. It is worth noting that given Ψ is known, our reconstruction method (term "Gaps sampler") **match the performance of an ideal benchmark for which the locations are known**. This result is particularly remarkable, as it demonstrates that our method is able to achieve the best possible performance. Indeed, observe that the reconstruction algorithm only takes Ψ (or a single number p_c for the ignorable case), as information about missing samples. Given an observation sequence O_i, there is an exponential number of possible placings of gaps within that sequence, and we know from the experiment described in Figure 1b that a random placement of gaps yield poor results. Yet, the algorithms manage either to get close to, or match the performance of the sampler that knows the locations of the missing values. To the best of our knowledge, this is the first time such results have been achieved.

Experiments and evaluations under miss-specifications. In Section 4.2, we evaluate the robustness of the algorithms with respect to various *model miss-specifications*. First we study the effect of providing a perturbed Ψ to the algorithm. Second, we study the ability to reconstruct when Ψ is unknown for some states. Third, we evaluate other miss-specifications w.r.t the omitting process: We tested the performance for a non-constant Ψ (that is, changing per sentence) and a scenario where the omitting process is a Markov process. For all of the above miss-specifications we show that the results are quite stable.

Code. To the best of our knowledge, our implementation is the first publicly available Gibbs sampling-based HMM learning implementation for Python, and the first to handle non-ignorable missing observations in general. The code is provided in the supplementary material and will be made publicly available with the final version of the paper.

2. Background and Model Definition

A Hidden Markov Model $(\mathbb{X}, T, \Theta)$ is defined by a finite state space $\mathbb{X}$ and a transition probability matrix $T \in R^{|\mathbb{X}|\times|\mathbb{X}|}$ which defines a Markov chain on $\mathbb{X}$. The observations of the model take values in a set $\mathcal{E}$, and for every state $X \in \mathbb{X}$, Θ_X is a distribution on $\mathcal{E}$ corresponding to X. Let $X_i = [X_{i0}, \ldots, X_{iN}]$ be a sequence of states, and let $U_i = [u_{i0}, \ldots, u_{iK}]$ be a sequence of observations (also referred to as *sentence*), $u_{ij} \in \mathcal{E}$, $K \leq N$.
The sequence W_i encodes the locations of the non-missing observations. Define $W_i = [w_0, \ldots, w_K]$ as a sequence of indexes $w_k \in \{0, .., N\}$,$w_k < w_{k+1}$, such that $w_k \in W_i$ if observation u_{ik} was generated at time w_k. We denote by $U_i^{W_i} = [u_{iw_0}, \ldots, u_{iw_K}]$ the restriction of U_i to W_i.

HMMOP - Unknown Gaps Model Definition.
Let $M' = (\mathbb{X}, T, \Theta)$ be an HMM. Define a random mapping $\Phi_\Psi(\cdot)$ of a full observation sequence U_i to a partial observation sequence O_i as follows: Given a full sequence of states, X_i, corresponding to an observation sequence, U_i, an indicator vector, $C_i = [c_{i0}, \ldots, c_{iN}]$ is sampled where c_{ij} are independent Bernoulli variables with success probability $\Psi(X_{ij})$. The set $W_i = \{j | c_{ij} = 1\}$ is defined, and the partial observation sequence, O_i, is set as $O_i = \Phi_\Psi(U_i) := U_i^{W_i}$. Then HMMOP is defined to be the generative model where a state sequence X_i and a full observation sequences U_i are generated by an HMM, and the corresponding observation sequences O_i are then produced as $O_i = \Phi_\Psi(U_i)$. The joint distribution of the HMMOP is :

$$P(O_i, W_i, X_i|\Theta) = P(X_0|\Theta)P(O_0|X_0) \cdot \prod_{t=1}^{N} T_{X_t, X_{t-1}} \cdot \begin{cases} (1 - \Psi(X_t)) \cdot P(O_t|X_t, \Theta), \text{if } t \in W_i \\ \Psi(X_t), \text{otherwise} \end{cases}$$

2.1. Non Ignorable Omitting Process

Non Ignorable omitting process in HMMs, also known as State-Dependent Missingness (SDM), was first introduced by (Yeh et al., 2012) and later developed by (Speekenbrink & Visser, 2021). Reconstruction with non-ignorable missing observations refers to the problem of reconstructing HMMs when the probability of an observation to be missing depends on the state of the system, in contrast to the ignorable case where $\Psi(s)$ is constant and simply translate to the percentage of missing observations in the data. Reconstruction of dynamical process with SDM (not necessarily HMMs) is considered a challenging problem and it is an active area of research. In this paper we generally consider the non-ignorable case and we carefully examine the impact of the SDM assumption on our reconstruction results. Notably, we show that we can still achieve ideal results for known omitting probabilities (Ψ) even in the SDM case. Section 4 presents comprehensive evaluation of the ignorable case.

2.2. Analytical Analysis

For analytical analysis we consider the case of ignorable missing observations, that is, $\Psi(s) = 1 - p_c \forall s$. More, let us first discuss the *non-hidden* case. That is, we assume for the moment that the observation given a state $X \in \mathbb{X}$ is the state X itself. Note that the unknown gaps reconstruction is interesting even in this simpler case.

Proposition 2.1. *Let $M' = (\mathbb{X}, T, \Theta)$ be an non-hidden Markov model. $\mathbb{I}$ the identity matrix. Then the sequence of observations $O_i = \Phi_{\Psi=p_c}(U_i)$ is a Markov chain with transition matrix :*

$$T_r = p_c \cdot T \cdot [\mathbb{I} - (1 - p_c) \cdot T]^{-1}. \tag{1}$$

Proof. By definition, in the non-hidden case the sequence U_i coincides with the state sequence X_i, and O_i is a subsequence of U_i. Note that the waiting time d between two occurrences of 1 in C_i is geometrically distributed with mean p_c, which implies that

$$T_r = \sum_{d=1}^{\infty} p_c(1 - p_c)^{d-1} T^d. \tag{2}$$

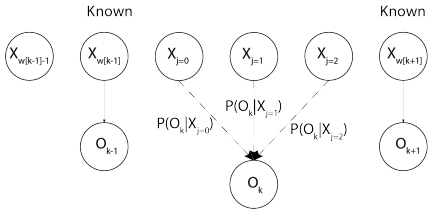
For every stochastic matrix T we have $\|T\| \leq 1$, (Goldberg, 1966), where $\|\cdot\|$ is the operator norm, $\|(1 - p_c) \cdot T\| < 1$, and thus the series in (2) are summable, with sum given by (1). □

Now, note that given the partial observations O_i, we can directly learn the transition matrix T_r for O_i by counting the co-occurrences of the states in O_i. This is the instantiation of the *Naive* approach. Next, observe that if p_c is known, using (1) we can *invert* the omission process to obtain T:

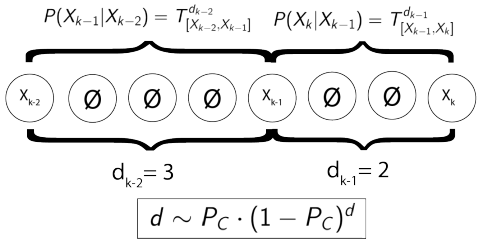
$$T = [\mathbb{I} \cdot p_c + (1 - p_c) \cdot T_r]^{-1} \cdot T_r. \tag{3}$$

We refer to this as the *backward transformation*.

As we can see, T is a transition matrix of a Markov model for any $p_c \in (0, 1]$. Therefore, given a set of missing observations O, it is not possible to infer any "original" T, but rather only a backward version of T that is conditioned on a specific value of p_c. As Supplementary material F shows, the difference between reconstructed T for different p_c might be significant.
The current approach to reconstructing Profile-HMMs involves guessing a random p_c, then using an Expectation-Maximization (EM) procedure to learn T given the guessed p_c, and vice-versa. However, as we demonstrated, guessing the p_c is equivalent to randomly selecting a backward transformation of T, which has no practical meaning. This we believe explain the lack of successful PHMMs reconstruction studies, despite its importance. Detailed evaluations of



(a) Illustration of W Sampling given $\Psi = p_c$.



(b) Illustration of HMMOP given $\Psi = p_c$.

Figure 2. Reconstruction methods.

(1),(3), together with additional results and characteristics of the backward transformation, are given in Supplementary Material Section F

Note that the "backward transformation" can be seen as a reconstruction method for the case of ignorable missing observations. **For the proper hidden case**, we can apply any standard HMM learning algorithm to learn T_r using the *Naive* approach from $\mathcal{O}$. Then, we can use (3) in the same way as in the non-hidden case to obtain T. Since this approach requires some HMM learning, which is typically non-analytic, we refer to this approach as semi-analytic. We evaluate the results of this method in compare to the other methods in Evaluation Section 4.

3. Reconstruction Methods

In this section, we present the reconstruction methods. As discussed in Sections 1.3 and 2.2, these methods rely on different levels of knowledge about the underlying latent dynamic process or the omitting process. **The first method**, named the "Matching Sampler," requires knowledge of the full unknown sequence length (N) as input. For the case of ignorable missing observations, this method does not require any additional information. However, in the case of non-ignorable missing observations, knowledge of the omitting probabilities (Ψ) is required. We show that even if this information is only partially known, it can be inferred as part of the algorithm and the method is highly robust to misspecifications regarding Ψ. **The second method**, named the "Gaps Sampler," only requires knowledge of Ψ as input. Although it does not have access to information about N, the "Gaps Sampler" is also somewhat robust to misspecifications regarding Ψ. Often, knowledge about N can be derived from prior knowledge on the underlying process M', for instance, when there are distinctive "start" and "end" states with knowledge about the latent process rate (Saelens et al., 2019; Herring et al., 2018). On the other hand, knowledge about Ψ exists for cases in which prior knowledge on the sampling method (Φ_Ψ) exists (Lummertz da Rocha et al., 2018). In addition, in cases where the reconstruction is a first step for a prediction task (Nishimoto et al., 2019), Ψ can be partially inferred as an hyper parameter using the labeled data (Supplementary Material Section E presents a prediction algorithm for HMMOP).

Gibbs sampling. Both of the methods relay on the Gibbs sampler (Gelman et al., 2013; Rydén, 2008). A Gibbs sampler samples the HMM parameters (i.e., T and Θ) and the latent state sequences $\mathcal{X}$ conditioned on $\mathcal{O}$. More specifically, one interchangeably samples $P(X_i|T, \Theta, \mathcal{O})$, and then samples $P(\Theta|X_i, \mathcal{O})$ and $P(T|X_i)$. Here we extend this procedure by incorporating the uncertainty of the unknown gaps into the sampling process using an additional set of latent helper variables. Let $\mathcal{X} = \{X_i\}$ be the set of all latent states sequences as describe above, and let $O_i \in \mathcal{O}$ be the observed sequences corresponding to X_i.

3.1. The Matching Sampler

The Matching Sampler builds upon $\mathcal{W} = \{W_i\}$ (see Section 2) as an helper set of latent variables, hence the challenge mainly lies in sampling the helper variables $P(W_i|X_i, N_i, O_i, T, \Theta)$. To address this challenge, we first note that the conditional distribution of W_i given X_i is independent of T. Therefore, our problem reduces to sampling $P(W_i = [w_{i0}, \ldots, w_{iK}]|\Theta, X_i, O_i, N_i, \Psi)$.
For $k \leq K$, let W^k denote the sequence W with w_k omitted. Also let $I_{[w_{k-1}, w_{k+1}]}(w)$ be the indciator function. The sampling of W_i will be done using its own Gibbs sampler. That is, instead of sampling W_i, we iteratively sample its components,w_{ik}, conditioned on the rest of the matching $P(w_{ik}|W_i^k, \Theta, X_i, O_i, \Psi)$. The Markov property of the sequence X implies that w_{ik} depends on the sequences W_i^k, O_i, X_i only through "local variables" at k, that is, it depends on $X_{i,w_{i,k-1}}$ up to $X_{i,w_{i,k+1}}$, and on the single observation O_{ik} :

$$\begin{aligned} &P(w_{ik}|W_i^k, \Theta, X_i, O_i, \Psi) \propto \\ &P(O_k|X_{w_k}) \cdot (1 - \Psi(X_{w_k})) \cdot I_{[w_{k-1}, w_{k+1}]}(w_k) \end{aligned} \tag{4}$$

This situation is illustrated in Figure 2a.

One can see a similarity between W_i to the "wrapping" vector from the known Stochastic Dynamic Time Wrapping (DTW) (Nakagawa & Nakanishi, 1988). Notice that while variations of SDTW can infer optimal W from $P(W|\ldots)$, our challenge is **sampling** $P(W|\ldots)$. Moreover, not every state in X_i is part of W_i as the SDTW algorithm requires.

In practice we use the Metropolis-Hasting (Metropolis et al.,

1953) for better convergence. Full details of the derivation of (4) and the complete algorithm, including the sampling of T, X, and W, as well as initialization considerations, can be found in Section C of the appendix.

3.2. The Gaps Sampler

The Gaps Sampler presents a different representation for the missing observations locations by introducing variables $d = [d_0, \ldots, d_K]$. Those variables represent the number of omitted observations between any two observed observations, which we denote as gaps. Figure 2b illustrates the Gaps sampler for the ignorable case for which $\Psi(\cdot) = p_c$. Our goal is then sample $P(d_k|X_{k,k+1})$. Lets define $s = [s_0, \ldots, s_{d-1}] \in S^d$ as the set of all d long states trajectories, also, lets assume $d < \mathcal{S}$ for a predefined $\mathcal{S}$. We say that a sequence is *fully omitted* if we didn't observed any of the sequence states, we denote the case of a fully omitted s as $\bar{s}$. Given Φ_Ψ independent, $P(\bar{s}|s) = \prod_{i=0}^{d-1} \Psi(s_i)$. By definition, $P(d_k|X_{k,k+1})$ is the probability for $s \in S^d$ been fully omitted:

$$P(d_k|X_{k,k+1}) =$$

$$\frac{\sum_{s\in S^d} T_{[X_k,s_0]} \cdot T_{[s_{d-1},X_{k+1}]} \cdot \Psi(s_0) \cdot \prod_{i=1}^{d-1} T_{[s_{i-1},s_i]} \cdot \Psi(s_i)}{\sum_{\tau=0}^{\mathcal{S}} \sum_{s'\in S^\tau} P(s'|X_{k,k+1}) \cdot P(\bar{s'}|s')}$$

$$\propto \sum_{s\in S^d} T_{[X_k,s_0]} \cdot T_{[s_{d-1},X_{k+1}]} \cdot \Psi(s_0) \cdot \prod_{i=1}^{d-1} T_{[s_{i-1},s_i]} \cdot \Psi(s_i)$$

And given that summing over $s \in S^d$ is not feasible, we suggest the following formulation:

$$= \sum_{x\in\mathcal{X}} \sum_{j\in\mathcal{X}} T_{[j,x]} \cdot \Psi(x) \cdot P(d-1|X_{k,k+1}, s_{d-2} = j)$$

$$P(d-1|(X_{k,k+1})) = \sum_{j\in\mathcal{X}} P(d-1|X_{k,k+1}, s_{d-2} = j)$$

This form the *forward algorithm* (Rabiner, 1989) with $P(d|X_{k,k+1}, s_{d-1} = j)$ as the forwarding elements. Notice that $P(\ldots, d-1)$ are intermediate steps for calculating $p(d)$, hence the computational complexity of $P(1), \ldots, P(\mathcal{S})$ and $P(\mathcal{S})$ are equal. Full details of the Gaps sampler are given in Section D of the appendix.

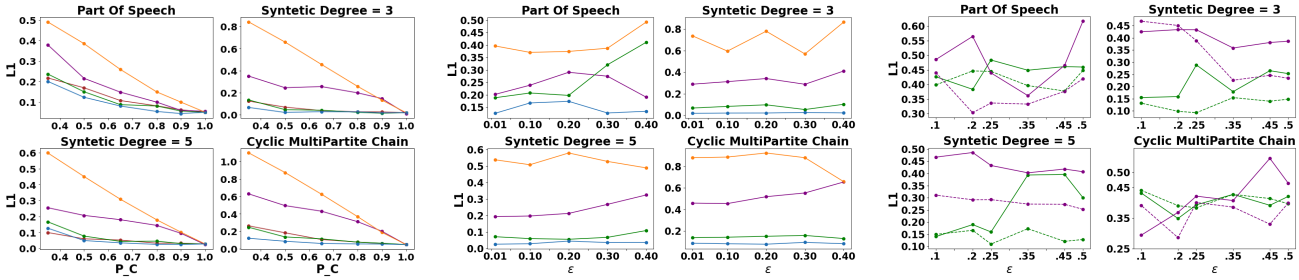
3.2.1. Sampling Ψ

Both methods include the ability to sample and impute Ψ for cases where Ψ is partially known. Note that $\Psi(s)$ follows a Bernoulli distribution, so the conjugate prior for $\Psi(s)$ for each s is the Beta distribution. Hence, the posterior distribution of $\Psi(s)$ given X and W is $\Psi(s)|X, W \sim Beta\left(\mu(s \in X^W), \mu(s \in X) - \mu(s \in X^W)\right)$ where $\mu(\cdot)$ is the counting measure. Section 4 extensively evaluate the ability of our methods to impute partial Ψ.

4. Experiments

This section evaluates the performance of the following reconstruction methods: 1. The Matching sampler (**Purple**); 2. The Gaps sampler (**Green**); 3. The *Naive* reconstruction as described in Section 1.2 (**Orange**); And, 4. The "ideal benchmark" (**Blue**) where the true gaps locations are known to the sampler as described in Section 1.1; Additionally, for the ignorable case, we evaluate the semi analytical approach (**Brown**) which is a byproduct of the analytical analyses;
The following models were used to generate the data (full details are given in Supplementary Material Section G): 1,2. **"Synthetic Degree$\{d\}$"** A synthetic chain on 10 states, where each state has d outgoing transitions, chosen at random; 3. **"Cyclic MultiPartite"** Synthetic chain with 25 states. The states are divided into 5 groups ranging from 1 to 5, transitions are only possible between states of consecutive groups and in a cyclic way. For example, a state from group 4 only moves to a state in group 0, etc; 4. **"Part Of Speech"** process. Transitions and emissions (part of speech and words respectively) probabilities were extracted from the Brown NLP corpus (Francis, 1965). The trajectories were then sampled based on those parameters.
With the exceptions of the "Part Of Speech" chains, state emissions are distributed with Normal distribution $N(\mu_i, 0.1)$ for $\mu_i \in [0, n_{state} - 1]$; Unless specified otherwise, 1500 sentence(U) of length 80(N) were sampled. For all figures, Y-axis is the L_1 distance between the reconstructed and the ground truth transition matrices. Supplementary Material Section H presents the standard deviations (s.d) of this section results.

Note that transitions are of paramount importance in biological settings, as the transition matrix often model the biological mechanism behind the process in question (Shokoohi et al., 2019; Ye et al., 2019; Yoon, 2009). Furthermore, in these instances, the states and the emissions probabilities are commonly known (Setty et al., 2019; Liu et al., 2017a; Lummertz da Rocha et al., 2018). In our experimental setup, we adopted parameters that bear close resemblance to those observed in biological systems. For instance, in numerous genomics applications employing HMMs, the state count is commonly 6 (comprising four observable and two hidden states) or 21, as elaborated in (Yoon, 2009). Moreover, it's worth highlighting that the transition matrix in such cases is often sparse. In addition, in Cell Trajectory Analyses (CTA), the state count is usually determined by user's discretion but typically does not exceed 20, as validated by (Shokoohi et al., 2019; Ye et al., 2019), where 12 states were deployed. In (Saelens et al., 2019), one of the most exhaustive CTA method comparisons to date, the maximum state count was



(a) Reconstruction for ignorable setup. (b) Reconstruction for nonignorable setup. (c) The impact of Ψ imputation.

Figure 3. **(a, c)** Both Samplers significantly improves on the *Naive* method. While the "Gaps" is better. **(b)** The Gap sampler converge to a better placement, faster.

set at 20. CTAs often can be seen as a branching process that incorporates "local" cycles and a slower "global" cycle, a presumption that is integrated into our bipartite model.

4.1. Reconstruction Under Correct Specification

This section evaluates the reconstruction performance of the methods with no miss-specification. Figure 3a presents the reconstruction results for the ignorable case, that is, only a single number is provided to the methods which is the true percentage of missing observations (or the true trajectory length N for the Matching sampler). In this experiment, we pick varying p_c's (X-axis), i.e. $\Psi = 1 - p_c$ percentage of observations are deleted. As the figure shows, the performance of all the algorithms is significantly better than the *Naive* algorithm, which for most cases, is the only applicable one (1.1). Moreover, for the Gaps sampler, the reconstruction performance **matches that of the ideal benchmark** (1.1). As discussed in Section 1.3, this result is remarkable, due to the exponential number of possible placements W.

To gain a better understanding of the performance distinctions between the samplers, we will conduct a more in-depth examination of them. To that end, we fix a ground-truth T, and learn the placement parameters W (or, equivalently, $\mathbf{d}$). That is, T is fixed, and in a single iteration of the sampler, we sample $P(X|W,T,O)$ and then $P(W|X,T,O)$. The performance was measured by evaluating the average log likelihood of the trajectories, $\frac{1}{|\mathcal{O}|}\sum_i^{\mathcal{O}} log\left(P(O_i, X_i|T, W_i)\right)$ after each iteration of the sampler. We have also varied the number of times $P(W|X,O)$ is sampled, that is, for a fixed X, the matching sampler had from 30 to 300 steps to find the "right" W, before the next X was sampled. For the Gaps sampler, only one step of W sampling is required by design per X sample. As shown in Figures 4, while on the degree $d = 5$ the performance is comparable, on the Cyclic MultiChain data the Gaps sampler finds more likely trajectories dramatically faster.

Figure 3b presents the evaluation results for the non-ignorable case. In this experiment, for each s we sample

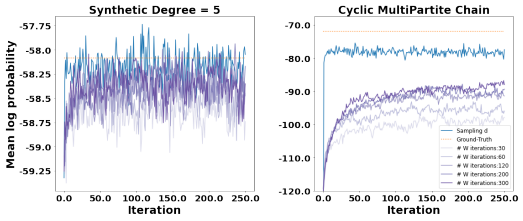


Figure 4. Convergence of the placements parameters.

$\Psi(s) \sim Uniform[0.5-\epsilon, 0.5+\epsilon]$ for varying ϵ (X-axis), which control the variance of $\Psi(s)$. After omitting observations according to $\Psi(s)$, we provide it to the methods. As the figure shows, the results are consistent with those of Figure 3a even for drastic variance in $\Psi(s)$. Previous works has demonstrated a decrease in performance when dealing with non-ignorable missing observations (Speekenbrink & Visser, 2021), but our methods appear to be effective in this case, likely due to the use of Gibbs sampling which impute missing observations as an intermediate step.

Figure 3c evaluates the performance for when only limited knowledge of Ψ is available. This experiment highlights a key advantages of sampling-based methods, which can infer missing values of Ψ. Specifically, we randomly generate a single $\Psi(s) \sim Uniform[0.35, 0.65]$, and then delete a proportion, ϵ, of entries. The partial $\Psi(s)$ is then provided to the methods. In the "Sampling imputation" case (dashed lines), the missing entries are inferred using the sampling procedure described in Section 3.2.1, while in the "Random imputation" case, they are filled in with random values.
As shown in Figure 3c, the imputation step leads to a significant improvement in performance. The Matching Sampler, which is provided with knowledge of N, demonstrates improved results across all benchmarks and levels of ϵ. Additionally, the Gaps Sampler, which does not have access to this information, also shows notable improvement for some benchmarks. These results are noteworthy, as knowledge of N is common in many applications of PHMMs, and obtaining prior knowledge about Ψ can be costly or difficult

to infer. Furthermore, the fact that the Gaps Sampler does not require knowledge of N, making this result even more impressive.

4.2. Robustness Under Misspecification

From a practical point of view, it is reasonable to assume that Ψ will be known to the reconstruction algorithms only up to some error. Therefore in this section we evaluate the robustness of the algorithms under different misspecifications. In view of the results of the previous section, in this section we focus on the "Gaps sampler". Additionally, we presents the results of the semi-analytical approach as a comparison to the ignorable case.

Figure 5a presents the case of "wrong p_c". Here we address the ignorable case where the p_c estimate provided to the algorithm differs from the ground truth value $p_c = \frac{1}{2}$. As the figure shows, **both of the algorithms are relatively robust with respect to wrong** p_c. Moreover, note that even for sizable error in p_c both algorithms are superior to the *Naive* approach. In addition, the sampler based algorithm tends to be better then the semi analytical algorithm.

Figure 5b presents the case of "non-constant" p_c, where there is no single constant ground truth p_c. Instead, for each sentence, p_c is sampled independently from a normal distribution $p_c \sim N(\frac{1}{2}, \sigma)$. In this case, the p_c provided to the algorithm is the average one ($p_c = \frac{1}{2}$). Notice that the larger the σ, the noisier the data is, and hence the harder the problem. Never the less, the Gaps sampler give great results even for high values of σ, outperforming the semi-analytic method in a more pronounced way. Given that the case of non constant p_c is the most realistic one, this result showcases the **benefit in sampling based methods**.

Robustness under Observations Removal Process Misspecifications. Previous sections assume that the omission process,Φ_Ψ (Section 2), is modeled as a series of independent Bernoulli trials. We now consider a case where instead the omission process is a Markov process with two states: $R_s =$ "$seen$" and $R_m =$ "$missing$", with transition probabilities given by $P(R_s|R_s) = p_c + \epsilon$ (and hence $P(R_m|R_s) = p_c - \epsilon$), and $P(R_s|R_m) = (1 - p_c) - \epsilon$ (and hence $P(R_m|R_m) = (1 - p_c) + \epsilon$). Thus, ϵ represents the bias of staying at the same state, with $\epsilon = 0$ case being equivalent to the i.i.d Brenoulli process above. Figure 5c compares the algorithms for different values of the bias ϵ. As the figure shows, the sampler results are mostly better than the ones of the *Naive* solution, and constantly better then the semi-analytical ones. Nevertheless, the results of the *Naive* solution improved as ϵ increases, and after some threshold of ϵ, the *Naive* algorithm become somewhat better then the sampler. As discussed in Proposition 2.1, the *Naive* reconstruction can be seen as the weighted mean of the ground-truth matrix T and wrong transition matrices T^d derived from sequences of length d of omitted observations only. As ϵ increases, the probability of longer consecutive sequences increases (that is, long sequences of $P(R_s|R_s)$ or $P(R_m|R_m)$). But, while longer sequences of consecutive observed states (R_s) are beneficial to the $Naive$ solution, the solution is negatively effected by the number of omitted observations sequences, and not their length.

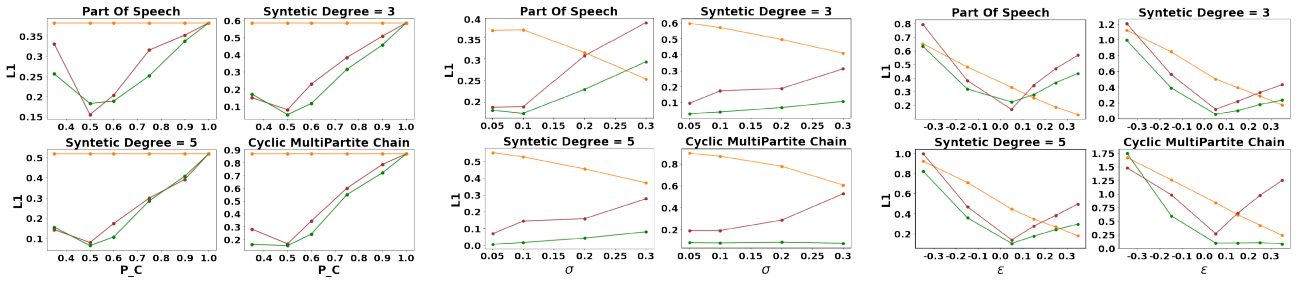
Robustness under Ignorability Assumption Misspecifications. Figure 6 examines the impact of considering the non-ignorable setup on the performance of the methods. For this, for each s we sample $\Psi(s) \sim Uniform[0.5 - \epsilon, 0.5 + \epsilon]$ for varying levels of ϵ (X-axis), and provide the algorithms with only the empirical p_c, or N for the Matching Sampler (dashed lines). We also compare the results to those of the *"ideal benchmark,"* which assumes fully observed missing locations but does not take into account the non-ignorability assumption. As the figure shows, while the performance is similar for low values of ϵ, it becomes more pronounced for higher variance for both the Gaps Sampler and the *"ideal benchmark."* This demonstrates the importance of considering the non-ignorability assumption, which is often overlooked in HMM reconstruction methods.

4.3. Computational Complexity of the Proposed Algorithms

The computational complexity of both the Gaps sampler and the Matching sampler depends on three factors: the complexity of sampling a single sentence, the number of of sentences, and the total number of iterations involved. When analyzing the complexity for a single sentence, the computational demands of both samplers are governed by the backward-forward sampling algorithm used for state sequence X sampling, and additionally, by the sampling of either missing location (W) or interval length assignments (D). The backward-forward sampling algorithm has a well-understood complexity of $O(2 \cdot S^2 \cdot N)$, where S symbolizes number of states and N represents the length of the latent process. Also, for each time point t, a sampling procedure from a Dirichlet distribution of size S is initiated.

Regarding to the sampling of interval lengths D, a forward algorithm is engaged (see Section 3.2), carrying a complexity of $O(S^2 \cdot N)$, complemented by a singular sampling operation from a Dirichlet distribution of size 100, as detailed in Section 3.2. The process of sampling W presents the most computation-intensive part of this paper, utilizing an additional inner sampler. Its complexity hinges on the product of the count of pre-determined sampling iterations and N. As depicted in Figure 4, the Gaps parameters achieve faster convergence, which consequently leads to quicker Gibbs convergence.

As a specific example, the most computationally demanding experiment conducted in this paper (featuring 1000 sen-



(a) Wrong $\mathbf{p_c}$. (b) Non-constant $\mathbf{p_c}$. (c) $\mathbf{\Phi_{p_c}}$ with memory.

Figure 5. Robustness Under Misspecification.

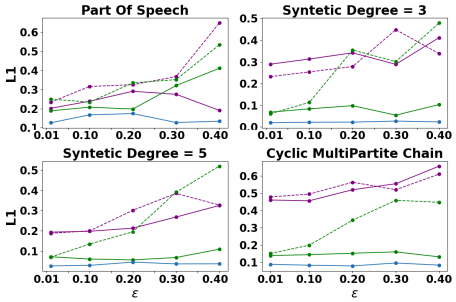


Figure 6. Assume ignorable for non-ignorable Φ.

tences of size 100, a matching sampler with 120 iterations for W learning, and the imputation of Ψ) would take an estimated 8 minutes. The majority of configurations, however, would require significantly less time. In our observations, this duration is considerably shorter than what the standard EM algorithm requires in the widely used Pomegranate package (Schreiber, 2016). The improvement in our computational performance is attributable to multiple elements, including efficient distribution sampling (especially Gaussian and Dirichlet) and full parallelism.

5. Conclusion

This study addresses the challenge of reconstructing hidden Markov models from ordered trajectories with missing observations at unknown locations, which is a common issue in fields such as computational biology. A novel and general approach was proposed, based on Gibbs sampler, which overcomes many of the limitations of existing methods and opens the way for new applications. Additionally, the approach addresses the non-ignorable missing observations setting . Moreover, the robustness of the algorithms to different misspecifications was demonstrated. Two notable results were presented: 1) the reconstruction performances are comparable to the performance of algorithms that use known missing locations, even in the non-ignorable case, and 2) partially known omission probabilities can be inferred from data. As future work, further improvement is possible by incorporating prior knowledge of the latent process structure into the Bayesian framework and by using specialized omitting process tailored to specific problems.

References

Campbell, K. R. and Yau, C. Uncovering pseudotemporal trajectories with covariates from single cell and bulk expression data. *Nature communications*, 9(1):1–12, 2018.

Chadza, T., Kyriakopoulos, K. G., and Lambotharan, S. Analysis of hidden markov model learning algorithms for the detection and prediction of multi-stage network attacks. *Future Gener. Comput. Syst.*, 108:636–649, July 2020.

Chen, J., Rénia, L., and Ginhoux, F. Constructing cell lineages from single-cell transcriptomes. *Mol Aspects Med*, 59:95–113, 02 2018.

Chib, S. Calculating posterior distributions and modal estimates in markov mixture models. *Journal of Econometrics*, 75(1):79–97, 1996. ISSN 0304-4076. doi: https://doi.org/10.1016/0304-4076(95)01770-4. URL `https://www.sciencedirect.com/science/article/pii/0304407695017704`.

Deconinck, L., Cannoodt, R., Saelens, W., Deplancke, B., and Saeys, Y. Recent advances in trajectory inference from single-cell omics data. *Current Opinion in Systems Biology*, 2021.

Eddy, S. R. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, 10 1998. ISSN 1367-4803. doi: 10.1093/bioinformatics/14.9.755. URL `https://doi.org/10.1093/bioinformatics/14.9.755`.

Finn, R. D., Clements, J., and Eddy, S. R. Hmmer web server, interactive sequence similarity searching. *Nucleic Acids Research*, 39, 05 2011.

Francis, W. N. A standard corpus of edited present-day american english. *College English*, 26(4):267–273, 1965. ISSN 00100994.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013. ISBN 9781439840955. URL https://books.google.co.il/books?id=ZXL6AQAAQBAJ.

Goldberg, K. Upper bounds for the determinant of a row stochastic matrix, 1966.

Hamilton, J. D. *Time series analysis*. Princeton university press, 2020.

Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57: 97–109, 1970. ISSN 0006-3444. doi: 10.1093/biomet/57.1.97.

Herring, C. A., Banerjee, A., McKinley, E. T., Simmons, A. J., Ping, J., Roland, J. T., Franklin, J. L., Liu, Q., Gerdes, M. J., Coffey, R. J., and Lau, K. S. Unsupervised trajectory analysis of single-cell rna-seq and imaging data reveals alternative tuft cell origins in the gut. *Cell Systems*, 6(1):37–51.e9, Jan 2018. ISSN 2405-4712. doi: 10.1016/j.cels.2017.10.012. URL https://doi.org/10.1016/j.cels.2017.10.012.

Higham, N. J. and Lin, L. On pth roots of stochastic matrices. *Linear Algebra and its Applications*, 435(3):448–463, 2011. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2010.04.007. URL https://www.sciencedirect.com/science/article/pii/S0024379510001849. Special Issue: Dedication to Pete Stewart on the occasion of his 70th birthday.

Kaufmann, S. *Hidden Markov models in time series, with applications in economics*. Chapman and Hall/CRC, 2019.

Liu, Z., Lou, H., Xie, K., Wang, H., Chen, N., Aparicio, O. M., Zhang, M. Q., Jiang, R., and Chen, T. Reconstructing cell cycle pseudo time-series via single-cell transcriptome data. *Nature Communications*, 8(1):22, June 2017a.

Liu, Z., Lou, H., Xie, K., Wang, H., Chen, N., Aparicio, O. M., Zhang, M. Q., Jiang, R., and Chen, T. Reconstructing cell cycle pseudo time-series via single-cell transcriptome data. *Nature communications*, 8(1):1–9, 2017b.

Lummertz da Rocha, E., Rowe, R. G., Lundin, V., Malleshaiah, M., Jha, D. K., Rambo, C. R., Li, H., North, T. E., Collins, J. J., and Daley, G. Q. Reconstruction of complex single-cell trajectories using cellrouter. *Nature Communications*, 9(1):892, Mar 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-03214-y. URL https://doi.org/10.1038/s41467-018-03214-y.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. doi: 10.1063/1.1699114. URL https://doi.org/10.1063/1.1699114.

Moor, M., Horn, M., Rieck, B., Roqueiro, D., and Borgwardt, K. Early recognition of sepsis with gaussian process temporal convolutional networks and dynamic time warping. In Doshi-Velez, F., Fackler, J., Jung, K., Kale, D., Ranganath, R., Wallace, B., and Wiens, J. (eds.), *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106 of *Proceedings of Machine Learning Research*, pp. 2–26. PMLR, 09–10 Aug 2019. URL https://proceedings.mlr.press/v106/moor19a.html.

Morimura, T., Osogami, T., and Ide, T. Solving inverse problem of markov chain with partial observations. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/32b30a250abd6331e03a2a1f16466346-Paper.pdf.

Murphy, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Nakagawa, S. and Nakanishi, H. Speaker-independent english consonant and japanese word recognition by a stochastic dynamic time warping method. *IETE Journal of Research*, 34(1):87–95, 1988. doi: 10.1080/03772063.1988.11436710. URL https://doi.org/10.1080/03772063.1988.11436710.

Nishimoto, S., Tokuoka, Y., Yamada, T. G., Hiroi, N. F., and Funahashi, A. Predicting the future direction of cell movement with convolutional neural networks. *PLOS ONE*, 14(9):1–14, 09 2019. doi: 10.1371/journal.pone.0221245. URL https://doi.org/10.1371/journal.pone.0221245.

Orr, J. W., Tadepalli, P., Doppa, J. R., Fern, X., and Dietterich, T. G. Learning scripts as hidden markov models, 2018.

Pattabiraman, S. and Warnow, T. Profile hidden markov models are not identifiable. *IEEE/ACM Trans Comput Biol Bioinform*, 18(1):162–172, February 2021.

Rabiner, L. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989. doi: 10.1109/5.18626.

Ramati.M. Irreguler-time markov model. Master's thesis, Ben Gurion University, Negev, 2010.

Rydén, T. EM versus Markov chain Monte Carlo for estimation of hidden Markov models: a computational perspective. *Bayesian Analysis*, 3(4):659 – 688, 2008. doi: 10.1214/08-BA326. URL https://doi.org/10.1214/08-BA326.

Saelens, W., Cannoodt, R., Todorov, H., and Saeys, Y. A comparison of single-cell trajectory inference methods. *Nature Biotechnology*, 37(5):547–554, May 2019. ISSN 1546-1696. doi: 10.1038/s41587-019-0071-9. URL https://doi.org/10.1038/s41587-019-0071-9.

Schreiber, J. pomegranate. *GitHub repository*, 2016.

Setty, M., Kiseliovas, V., Levine, J., Gayoso, A., Mazutis, L., and Pe'er, D. Characterization of cell fate probabilities in single-cell data with palantir. *Nature biotechnology*, 37(4):451–460, 2019.

Shokoohi, F., Stephens, D. A., Bourque, G., Pastinen, T., Greenwood, C. M., and Labbe, A. A hidden markov model for identifying differentially methylated sites in bisulfite sequencing data. *Biometrics*, 75(1):210–221, 2019.

Speekenbrink, M. and Visser, I. Ignorable and non-ignorable missing data in hidden markov models, 2021. URL https://arxiv.org/abs/2109.02770.

Van den Berge, K., De Bezieux, H. R., Street, K., Saelens, W., Cannoodt, R., Saeys, Y., Dudoit, S., and Clement, L. Trajectory-based differential expression analysis for single-cell sequencing data. *Nature communications*, 11 (1):1–13, 2020.

Wheeler, T. J. and Eddy, S. R. nhmmer: DNA homology search with profile HMMs. *Bioinformatics*, 29(19):2487–2489, July 2013. doi: 10.1093/bioinformatics/btt403. URL https://doi.org/10.1093/bioinformatics/btt403.

Ye, Y., Gao, L., and Zhang, S. Circular trajectory reconstruction uncovers cell-cycle progression and regulatory dynamics from single-cell hi-c maps. *Advanced Science*, 6(23):1900986, 2019.

Yeh, H.-W., Chan, W., and Symanski, E. Intermittent missing observations in discrete-time hidden markov models. *Communications in Statistics - Simulation and Computation*, 41(2):167–181, 2012. doi: 10.1080/03610918.2011.581778. URL https://doi.org/10.1080/03610918.2011.581778.

Yoon, B.-J. Hidden markov models and their applications in biological sequence analysis. *Current genomics*, 10(6): 402–415, 2009.

Yu, D. and Deng, L. *Automatic Speech Recognition*. Springer, 2016.

A. Naive Reconstruction and Sensitivity w.r.t W Placement

Figure 7 presents the performance of the *Naive* method for additional models.
Figure 8 demonstrate the effects of gap location under different perturbation scenarios. *The first* perturbation (named "Equivalent") assumes the new locations points to the same states in the original sentence, For example, for sentence [A,B,B,C] the locations vector (0,1,3) and (0,2,3) are equivalent. *The second* permutation (named "Consecutive") assumes that the consecutive observations(observations without gaps between them) locations are preserved and the number of consecutive observations is the same. For example, if the non-gaps locations are (1,2,5,7,8,10) we create (1,2,4,7,8,16). As the experiments shows, ignoring the missing observations or randomizing them gives quite bad results, although, the reconstruction not necessarily relays on the exact gaps as shown in the "equivalent" case.

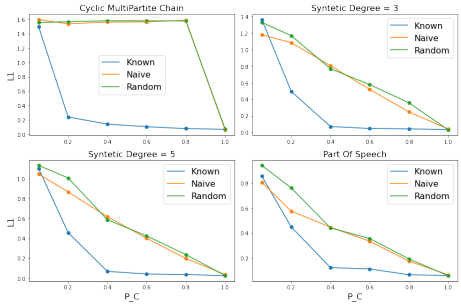


Figure 7. Reconstruction with the *Naive* model and with random allocations of missing observations locations.

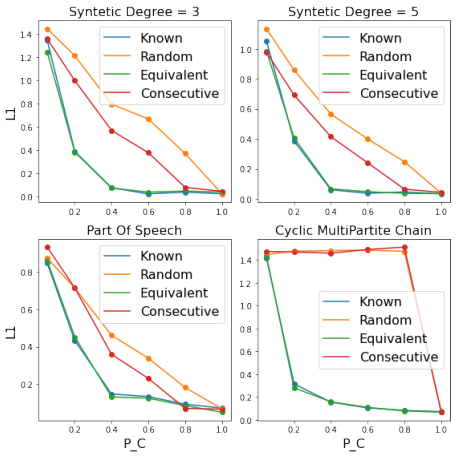


Figure 8. Reconstruction under different permutations of missing observation locations. X-axis is the p_c of Phi_{p_c}, Y-axis is the L1 distance from the original T.

B. Handling missing observations in HMM

Missing observations are an integral part in the practical use of HMMs. This section presents a general approach of handling missing observation in the case of known locations.

Lets define a simple discrete Markov model where $O_{1:N} = (O_1, ..., O_N)$ denote a time series of observations, and let Θ denote a vector of parameters. A hidden Markov model associates observations with a time series of hidden (or latent) discrete states $X_{1:N} = (X_1, ..., X_N)$. $T_{X_t, X_{t-1}} = p(X_t|X_{t-1})$ is the transitions probabilities between states. The joint distribution of observations and states can be stated as

$$P(O_{0:T}, X_{0:T}, \Theta) = P(X_0|\Theta)P(O_0|X_0, \Theta)\prod_{t=1}^{T} T_{X_t, X_{t-1}} \cdot P(O_t|X_t, \Theta) \tag{5}$$

Now let us assume some of the observations $O_{1:T}$ are missing. O^{miss} will be the set of all missing observations and O^{obs} the set for all observed observations. We can redefine as follow :

$$P(O_t|X_t,\Theta) = \mathbb{I}_{O_t \in O^{obs}} P(O_t|X_t,\Theta) + \mathbb{I}_{O_t \in O^{miss}} P(O_t = \emptyset|X_t,\Theta) \tag{6}$$

In our case the process removing the observations is dependent of Θ,X or O (non-ignorable) :

$$= \mathbb{I}_{O_t \in O^{obs}} P(O_t|X_t,\Theta) \cdot (1 - \Psi(X_t)) + \mathbb{I}_{O_t \in O^{miss}} \Psi(X_t) \tag{7}$$

Hence, if we define $W[w_1, w_2, ...]$ as a mapping between the $O_t \in O^{obs}$ to the corresponding time t (as in man paper), we can write:

$$\begin{aligned} &P(X,T,\Theta|W,\mathcal{O},\Psi) = \\ &P(X_0|\Theta)P(O_0|X_0,\Theta) \cdot \prod_{t=1}^{N} T_{X_t,X_{t-1}} \cdot \begin{cases} (1-\Psi(X_t)) \cdot P(O_t|X_t,\Theta), & \text{if } t \in W_i \\ \Psi(X_t), & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

C. Matching Sampler

This section contains a fully detailed Gibbs sampler for the case of reconstructing HMMOPs with known N. The section begin with addressing the initialization of the sampler for HMMOPs reconstruction, proceed to describe the "known location" Gibbs sampler while assuming W is known, and finally described the process of sampling W with more details.

Since the precise nature of emissions is not critical for the general results, from now on we will concentrate on the case of HMM with scalar Gaussian emissions, $\Theta_{x_n} = N(\mu_n, \sigma)$, where μ_n is a learnable parameter.

We starts with presenting the parameters of our model :

- T is the matrix which holds the probabilities to move from state X_i to X_j, with a Dirichlet distribution prior.
- $\mathcal{X}$ is the set of all latent states sequence X_i.
- $\mathcal{W}$ is the set of all mapping W_i corresponding to X_i, O_i.

The external parameters of the model are :

- $\{N_i\}$ the expected length of each $|U_i|$ (or $|X_i|$ equivalently) .
- σ_j the standard deviation(S.D) of the distributions of X's emissions.
- $\{O_i\}$ Our input (Missing sentences).
- Ψ are the omitting probabilities.

For simplicity, given that we sample each W_i or X_i independently, this section proceed with the notations - $O = [o_0, \ldots, o_K] := O_i$, $W = [w_0, \ldots, w_K] := W_i$, $X = [x_0, \ldots, x_N] := X_i$. Also, we define W^k as the sequence W when the w_k is omitted, and $X^{[a,b]}$, $O^{[a,b]}$ as the sub-sequence of X, O starting with index a and end with b.

C.0.1. Calculating Initial Values

(Rydén, 2008) offers the following steps for drawing initial conditions for μ given Normal prior and where $\hat{O}_i$ is the "full observed trajectory" (U_i, not exist in our case)- Each μ_i is given an independent Normal prior $N(\xi, \kappa^{-1})$ with $\xi = \frac{min_t(\hat{O}_{i,t}) + max_t(\hat{O}_{i,t})}{2}$ and $\kappa = \frac{1}{R^2}, R = max_t(\hat{O}_{i,t}) - min_t(\hat{O}_{i,t})$. Because we cant map observations to times without W, we cant directly draw the initial conditions as in the simple case. So, we used an unsupervised approach to asses

the mapping between observations and distributions. Given the prior of Normally distributed states, our observations are a set of samples drawn from the GMM $\sum_l^{|D|} d_l$, so for assigning observations to distributions we used an EM algorithm for GMM . Then, for $\{O_l\}$ all the observations assigned to distribution d_l, we calculate $\xi_l = \frac{min(O_k)+max(O_k)}{2}$ and $\kappa_l = \frac{1}{R^2}, R = max(O_k) - min_t(O_k)$.
Initial conditions for T and W are drawn in random .

C.0.2. Sampling Conditional Distributions for known W

Gibbs sampler is a common algorithm for HMM reconstruction, which allow us to draw samples from the posterior distribution of the HMMs (or in our case 8). Here we will follow the work of (Rydén, 2008). The idea beyond Gibbs sampling is alternating between sampling model parameters and latent data from their respective full conditional distributions. This is because, given the latent Markov chain and the data, the parameters are conditionally independent. And vice versa, given the parameters and the data the latent process is a non-homogeneous Markov chain and hence simple to sample. The basic steps for sampling the posterior $P(T, X, \Theta, W|\mathcal{O}, N, \Psi)$ for the reconstruction are 1. start with initial values for the parameters 2.sample a latent sequence X form the posterior given the parameters 3. sample the parameters form their conditional distribution, X and the observations.

Sampling T.
As we defined earlier, T the transitions probabilities are assumed to be drawn from an NXN Dirichlet distribution, so for sampling T we sample the distribution :

$$T|X, W, \mu = T|X \sim Dir(\sum_i n_{i1} + 1, \sum_i n_{i2} + 1, ..., \sum_i n_{id} + 1) \tag{9}$$

where n_{ij} is the number of transitions from state i to state j over all X .

Sampling μ.
Notice that for sampling μ, T is no longer needed. μ_x, the mean of the emissions distributions of state x, deepens on the observations correspond to X, hence, depends on both X and W. From (Rydén, 2008) we know :

$$\mu|... \sim N(\hat{\mu}, \hat{\sigma}); \hat{\mu} = (S_i + \kappa\xi)/(n_i + \kappa); \hat{\sigma} = 1/(n_i + \kappa) \tag{10}$$

Where:

$$S_n = \sum_{i,k} O_{ik} \cdot \delta_{X_{iW_k}, d_n}$$

[1]

Sampling X.
Given Θ,T and O_i, $X|\ldots$ can be modeled as a (non-homogeneous) Markov chain with initial distribution:

$$P(X_0|...) \propto \widehat{N}(O_0|X_0) \cdot P(O^{[2,n]}|X_0) \; . \; \widehat{N}(O_{it|X_j}) = \begin{cases} (1 - \Psi(X_t)) \cdot N(O_{it}|X_j) & if\ t \in W_i \\ \Psi(X_j) & else \end{cases} \tag{11}$$

and transition probabilities:

$$P(X_k|X_{k-1}) \propto T_{X_k, X_{k-1}} \cdot \widehat{N}(O_k|X_k) \cdot P(O^{[k+1,N]}|X_k)$$

For sampling X, we use an approach similar to the "forward backward" algorithm, called the "backward recursion forward sampling" algorithm (Chib, 1996). The algorithm sample X from the Markov chain similarly to how F-B algorithm sample observations from an HMM.

Finally, Our sampling algorithm is presented in algorithm 1

[1] δ_{xy} is the Kronecker Delta

Algorithm 1 Gibbs sampler given N

infer ξ and κ from $\mathcal{O}$ and calculate μ
Build random T and W
sample $X|\mathcal{O}, T, W, \mu$.
repeat
  sample $\mu|X, \mathcal{O}, W, \kappa, \xi, \sigma$.
  sample $T|X$.
  sample $W|X, \mathcal{O}, \mu, \sigma, \Psi$.
  sample $X|T, W, \mathcal{O}, \mu, \Psi$.
until convergence

C.0.3. SAMPLING W

Given the independence between the conditional distributions of each of the parameters in the Gibbs sampler, the difference between sampling $P(\mathcal{X}, T, \Theta, \mathcal{W}|\mathcal{O}, \Psi)$ instead of $P(\mathcal{X}, T, \Theta|\mathcal{W}, \mathcal{O}, \Psi)$ lay in sampling $P(\mathcal{W}|T, \mu, \mathcal{X}, \mathcal{O}, \Psi)$.

As described in Section 2, $\mathcal{W}$ are independent. Also, given X_i, T is no longer needed for the conditional distribution of W. So, our problem become sampling:

$$P(W = [w_0, \ldots, w_K]|\Theta, X, O, \Psi) \tag{12}$$

For sampling from (12) we used a Gibbs sampler once again, that is, instead of sampling W from (12), we iteratively sample w_k from :

$$P(w_k|W^k, \Theta, X, O, \Psi) \tag{13}$$

Given $w_{k-1} < w_k < w_{k+1}$, the probability (13) is of mapping a single observations O_k to one of $X^{[w_{k-1}, w_{k+1}]}$. And given the Markov property of X and that $Phi(\cdot)$ is a memoryless process, this mapping is independent from $w_{k'}, X_{k'}, O_{k'}$ for $k' \notin \{k-1, k, k+1\}$. So (13) become :

$$P\left(w_k \middle| O_k, X^{[w_{k-1}, w_{k+1}]}, \Psi\right) \cdot I_{[w_{k-1}, w_{k+1}]}(w_k)^2$$

$$= \frac{P(O_k|X^{[w_{k-1}, w_{k+1}]}, w_k)}{P(O_k|X^{[w_{k-1}, w_{k+1}]})} \cdot P(w_k|X^{[w_{k-1}, w_{k+1}]}) \cdot I_{[w_{k-1}, w_{k+1}]}(w_k)$$

$$= \frac{P(O_k|X_{w_k})}{P(O_k|X^{[w_{k-1}, w_{k+1}]})} \cdot \left(\frac{(1 - \Psi(X_{w_k}))}{\sum_j^{[w_{k-1}, w_{k+1}]}(1 - \Psi(X_j))}\right) \cdot I_{[w_{k-1}, w_{k+1}]}(w_k)$$

and the denominator is independent of w_k :

$$\propto P(O_k|X_{w_k}) \cdot (1 - \Psi(X_{w_k})) \cdot I_{[w_{k-1}, w_{k+1}]}(w_k) \tag{14}$$

C.0.4. M-H ALGORITHM

For better convergence we used a special case of Gibbs sampling called the M-H algorithm (Hastings, 1970). M-H is a special case of Gibbs sampling where the update from the new iteration W^{t+1} is conditioned with an *acceptance ratio* $\alpha = \frac{P(W^{t+1}|X_i, O_i)}{P(W_t|X_i, O_i)}$. Lets start with describing $P(W|O_i, X_i)$:

$$P(W|O_i, X_i) = \frac{P(O_i|W, X_i) \cdot P(W|X_i)}{P(O_i|X_i)} \tag{15}$$

For the ignorable case where Ψ is independent of X we have :

$$= \frac{P(O_i|W, X_i) \cdot p_c^{|O|} \cdot (1 - p_c)^{(|X| - |O|)}}{P(O_i|X_i)} \tag{16}$$

[2] $I_{[w_{k-1}, w_{k+1}]}(w_k)$ is the indicator function

And because $P(O_i|X_i)$ is independent of W :

$$\propto P(O_i|W, X_i) \cdot p_c^{|O|} \cdot (1-p_c)^{(|X|-|O|)} \tag{17}$$

Given $P(O_i|W, X_i)$ is simply the probability of observation sequence O_i on state sequence X_i with known mapping, we can say:

$$\alpha = \frac{p_c^{|O|} \cdot (1-p_c)^{(|X|-|O|)} \cdot \prod_{k=0}^{K-1} P(o_k|X_{iw_k^{t+1}})}{p_c^{|O|} \cdot (1-p_c)^{(|X|-|O|)} \cdot \prod_{k=0}^{K-1} P(o_k|X_{iw_k^{t}})} = \frac{\prod_{k=0}^{K-1} P(o_k|X_{iw_k^{t+1}})}{\prod_{k=0}^{K-1} P(o_k|X_{iw_k^{t}})} \tag{18}$$

For the non-ignorable case : Because $P(O_i|X_i)$ is independent of W, and given $P(O_i|W, X_i)$ is simply the probability of observation sequence O_i on state sequence X_i with known mapping :

$$\alpha = \frac{P(W^{t+1}|X_i) \cdot \prod_{k=0}^{K-1} P(o_k|X_{iw_k^{t+1}})}{P(W^{t}|X_i) \cdot \prod_{k=0}^{K-1} P(o_k|X_{iw_k^{t}})} \tag{19}$$

While $P(W^{t+1}|X_i), P(W^{t+1}|X_i)$ are hard to evaluate, the difference between them is only one mapping w. Lets assume w_j is the mapping which been updated in time t :

$$\alpha = \frac{(1-\Psi(X_{w_j^{t+1}})) \cdot \prod_{k=0}^{K-1} P(o_k|X_{iw_k^{t+1}})}{(1-\Psi(X_{w_j^{t}})) \cdot \prod_{k=0}^{K-1} P(o_k|X_{iw_k^{t}})} \tag{20}$$

The full sampling algorithm is described in Algorithm 2.

Algorithm 2 M-H sampler For W

Initial W^0 randomly
for i=0,1...,Number of iterations **do**
  $W^{t+1} = W^t$
  for k = 0,1,...,K **do**
    $W_k^{t+1} \sim P(w_k|w_{k-1}^{t+1}, w_{k+1}^{t})$
  end for
  calculate $\alpha = \frac{P(W^{t+1})}{P(W^t)}$
  sample u from uniform distribution over [0,1]
  if u $\leq \alpha$ **then**
    $W^t = W^{t+1}$
  else
    $W^{t+1} = W^t$
  end if
  $W^t = W^{t+1}$
end for

D. Gap sampler

Lets $M = (\mathbb{X}, T, \Theta)$ be an HMMOP, and $M_d = (\mathbb{X}, T^d, \Theta)$ be an HMM where T^d is the d-step transition matrix of transition matrix T. $D_\Psi(X_i, X_{i+1})$ is a random variable distributed according to the distances between two observable states X_i, X_{i+1}. From now on, we define $d_i = [d_{i0}, \ldots, d_{iK}]$ as the sequence of intervals. Note that d_{ik} is corresponding to

$w_{k+1} - w_k$ in the previous notation, as the number of gaps between two observed states in the original full states sequence U_i. $\mathcal{D}$ is the set of all d_i. As before, we omit the index i given the formulation is the same between sequences.

When conditioning over d, the posterior distribution of the HMMOP can be written as :

$$P(X, T, \Theta | d, O, \Psi) = P(X_0|\Theta, d_0) \cdot P(O_0|X_0, \Theta) \cdot \prod_{k}^{K_i} P(X_{k+1}|X_k, d_k) \cdot P(O_{k+1}|X_{k+1}, \Theta) \tag{21}$$

Samplind d : Our goal is to calculate :

$$P(d_k|(X_k, X_{k+1}))$$

Lets define $s = [s_0, \ldots, s_{d-1}] \in S^d$ as the set of all d long states trajectories. Also lets assume d is limited to be no longer then a predefined number $\mathcal{S}$. We say that a sequence is *fully omitted* if we didn't observed any state from the sequence. We denote the case of a fully omitted s as $\bar{s}$. The probability of s to be $\bar{s}$:

$$P(\bar{s}|s) = \prod_{i=0}^{d-1} \Psi(s_i)$$

By definition, $P(d_k|(X_k, X_{k+1}))$ is the probability for a sequence of length d (hence $s \in S^d$), been fully omitted, between (X_k, X_{k+1}).

$$\begin{gathered} \frac{\sum_{s \in S^d} P(s|(X_k, X_{k+1})) \cdot P(\bar{s}|s)}{\sum_{\tau=0}^{\mathcal{S}} \sum_{s' \in S^\tau} P(s'|(X_k, X_{k+1})) \cdot P(\bar{s'}|s')} = \\ \frac{\sum_{s \in S^d} T[X_k, s_0] \cdot T[s_{d-1}, X_{k+1}] \cdot \Psi(s_0) \cdot \Pi_{i=1}^{d-1} T[s_{i-1}, s_i] \cdot \Psi(s_i)}{\sum_{\tau=0}^{\mathcal{S}} \sum_{s' \in S^\tau} P(s'|(X_k, X_{k+1})) \cdot P(\bar{s'}|s')} \end{gathered} \tag{22}$$

Given the limited number of d's, we only need the probability up to proportion :

$$\propto \sum_{s \in S^d} T[X_k, s_0] \cdot T[s_{d-1}, X_{k+1}] \cdot \Psi(s_0) \cdot \Pi_{i=1}^{d-1} T[s_{i-1}, s_i] \cdot \Psi(s_i) \tag{23}$$

Lets present the case of $\Psi(\cdot) = (1 - p_c)$:

$$\begin{aligned} &= \sum_{s \in S^d} T[X_k, s_0] \cdot T[s_{d-1}, X_{k+1}] \cdot (1 - p_c) \cdot \Pi_{i=1}^{d-1} T[s_{i-1}, s_i] \cdot (1 - p_c) \\ &= (1 - p_c)^{d-1} \cdot \sum_{s \in S^d} T[X_k, s_0] \cdot \Pi_{i=1}^{d-1} T[s_{i-1}, s_i] \cdot T[s_{d-1}, X_{k+1}] \\ &= (1 - p_c)^{d-1} \cdot T^d[X_k, X_{k+1}] \end{aligned} \tag{24}$$

Back to the general case, we evaluate 23 using the forward algorithm. We can say that :

$$P(d|(X_k, X_{k+1})) = \sum_{x \in \mathcal{X}} \sum_{j \in \mathcal{X}} P(d-1|(X_k, X_{k+1}), s_{d-2} = j) \cdot T[j, x] \cdot (1 - \Psi(x))$$

While :

$$P(d-1|(X_k, X_{k+1})) = \sum_{j \in \mathcal{X}} P(d-1|(X_k, X_{k+1}), s_{d-2} = j)$$

As we can see this is the form of the forward algorithm with $P(d|(X_k, X_{k+1}), s_{d-1} = j)$ as the forwarding elements. Notice that $P(1, \ldots, d-1)$ are intermediate steps for calculating $p(d)$ in the forward algorithm, hence the complexity of calculating $P(1), \ldots, P(\mathcal{S})$ is equal to the complexity of calculating $P(\mathcal{S})$.

Algorithm 3 Gibbs sampler given PC

infer ξ and κ from $\mathcal{O}$ and calculate μ
sample T from a uniform Dirichlet prior
sample $X_{walk}|\mathcal{O}, T, \mu$.(naive sampling given T)
sample $d|T, X_{walk}, p_c$
repeat
  sample $\mu|X_{walk}, \mathcal{O}, \kappa, \xi, \sigma$
  build W,N from d
  sample $X_T|W, N, T, \mathcal{O}, \mu, \Psi$
  sample $T|X_T$
  sample $X_{walk}|T, d, \mathcal{O}, \mu, \Psi$
  sample $d|T, X_{walk}, \Psi$
until convergence

Sampling X : Given the equivalence between $\{d\}$ to $\{w\}$, we sample X in the same way as with the Matching sampler.

Sampling T : A challenge in the new representation is sampling $T|X, d, \Psi$. Notice that even in the ignorable case, X contains samples from T^d rather than T, and because of the difficulty in finding roots for stochastic matrices (Higham & Lin, 2011) and the sparsity of most of T^d, it is hard to calculate T^d explicitly in order to sample T. For those reasons, we use the fact that $w_k = \sum_{j=0}^{j=k} d_j$ and $N = \sum d_k$ and proceed as with sampling $P(T|X, \Theta, W, O, N)$.

Algorithm 3 describes the full Gibbs sampler.

E. Inference

A Common use of Hmm is sequence labeling (or inference, interchangeably). That is, given observations sequence O and an HMM with known parameters we aim to find $X_{ml} = argmax_X\ P(X|W, O)$. This section provide a inference method for sequence drawn from a HMMOP, so our aim is to find :

$$X_{ml} = argmax_X\ P(X, W|O) \tag{25}$$

Notice, that this case refer to a scenario where T from the fully observed HMM is known, but, the sentences have been drawn from the corresponding HMMOP. Given HMMOP is an HMM (2.1), the best representation for the dynamics of the labels drawn from the HMMOP is T_r. So, this case reefer to a scenario where T_r cannot be learn, for example because of lack of training data, but the "original" T is available.

Many algorithms exists for the case of known W. So, for solving (25) we use an iterative Expectation-maximization (E-M) algorithm based on the sub-problem :

$$W_{ml} = argmax_W\ P(W|X, O) \tag{26}$$

As before, $W = [W_0, \ldots, W_K]$ are mappings between $O_k \in O$ to $X_n \in X$. So, our goal is to find:

$$\begin{gathered} argmax_W \prod_{k=0}^{K-1} P(O_k|X_{W_k}) = argmax_W \sum_{k=0}^{K-1} log(P(O_k|X_{W_k})) \\ W_{k-1} < W_k < W_{k+1} \end{gathered} \tag{27}$$

This problem can be represented as finding longest path on a directed acyclic graph (DAG). Lets define a DAG $G = (S, T)$ where $s_{a,b} \in S, a \in [0, K-1], b \in [0, N-1]$ are the nodes with $r_{a,b}$ as weights and T' as the binary matrix representing the edges :

$$r_{a,b} = \begin{cases} log(P(O_a|X_b)) & \text{,if } b \geq k \\ -\infty\ else \end{cases} \qquad T[s_{a_i,b_i}, s_{a_j,b_j}] = \begin{cases} 1 & \text{,if } (a_j - a_i) == 1\ \ \&\ \ b_j > b_i \\ -\infty\ else \end{cases} \tag{28}$$

Now lets consider the problem of finding the longest path $V' = [s'_{a_o,b_o}, \ldots, s'_{a_K,b_k}]$ on G that starts in one of $s_{0,:}$ and ends in one of $s_{:,N}$. Given T', only transitions with consecutive a are possible, so, $|V'| = K$. Also, V' is ordered for b, so V' is a

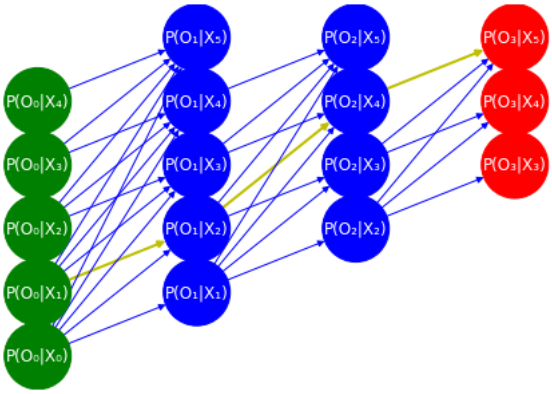


Figure 9. We want to find the longest path that start in green node and end in red node. In this case. Here $[log(P(O_0|X_1)), log(P(O_1|X_2)), log(P(O_2|X_4)), log(P(O_3|X_5))]$ so $W_{ml} = [1, 2, 4, 5]$.

sequence maximizing a sum of weights (28), where a are consecutive and b are ordered. In other words, by taking b for each $s' \in V'$ we get the optimal W. Figure 9 present an illustration of the process.

For better initialization, we calculated an initial W_{ml} instead of using random allocation. Given T, $P(d_k = n|X_k, X_{k+1})$ is given by **??**. Our goal is to solve :

$$\begin{gathered} argmax_d \prod_{k=0}^{K} P(d_k|X_k, X_{k+1}) = argmax_d \sum_{k=0}^{K} log(P(d_k|X_k, X_{k+1})) \\ d_{K+1} + \sum d_k < N \end{gathered} \tag{29}$$

where d_{K+1} is the probability for d gaps from X_K to end. This problem is known as the multi-choice knapsack problem where one pick a single item d_k from 1,...,N-K per k, that maximizes a "benefit" ($\sum_k P(d_k)$) with constrain over some "cost" ($\sum d_k < N$). The details of the multi-choice knapsack algorithm are in the code.

E.1. Inference results - Sequence Labeling

As presented in E, the reference algorithm receive 3 inputs - 1) a transition matrix T (derived from a fully observed HMM). 2) missing sentences 3) assumption for the percentage of missing observations, in our case the original sentence length N. Than, the algorithm returns a sequence of states of the length of the observations sequence.

For this experiment we sampled sentences from the models in G and remove observation according to $p_c = .5$. Than, we predict the labels of each sentence and compare them to the known ones. The evaluation measures are the mean and variance of the accuracy across sentences. We compare the results of five algorithms: 1) naive prediction on the full sentences (named "Full sentence"). 2) Based on the emissions probabilities only ("Emissions only"). 3) Missing sentences when the gaps locations are known ("Naive"). 4) missing sentences using a naive prediction. 5) missing sentences using the HMMOP inference with known N("HMMOP").

Given the sensitivity of the label prediction task to the emissions probabilities, for the Gaussian models we compare the results for different σs. As Table 1 present, the HMMOP give better results under all scenarios. Also, its interesting to see that when ignoring the missing observations, one better relay on the emissions alone rather using the wrong dynamical data.

Table 1. Label prediction results

	Synthetic Degree = 5			Synthetic Degree = 3		
STD	0.5	0.75	1.0	0.5	0.75	1.0
Full sentence	.87(.08)	.80(.11)	.64(.10)	.96(.05)	.98(.05)	.82(5.6)
Emissions only	.69(.15)	.57(.15)	.42(.17)	.51(.15)	.72(.13)	.43(5.6)
Known W	.79(.15)	.65(.16)	.58(.17)	.90(.12)	.89(.10)	.60(5.6)
Naive	.66(.15)	.52(.17)	.49(.18)	.80(.21)	.49(.18)	.38(5.6)
HMMOP	.73(.14)	.59(.14)	.50(.17)	.84(.16)	.73(.15)	.45(5.6)

Multi-Partite			POS
0.5	1.0	1.5	
.91(.06)	.78(.09)	.69(.12)	.91(.06)
.71(.15)	.50(.15)	.40(.16)	.89(.11)
.83(.13)	.64(.17)	.57(.18)	.90(.10)
.69(.16)	.48(.18)	.40(.17)	.88(.11)
.72(.16)	.50(.16)	.40(.15)	.89(.12)

F. Analytical Reconstruction - Ignorable Case

Given $OP(\cdot)$, the *Naive* algorithm reconstruct $T_{missing}$ which can be described as follow:

$$\begin{aligned} T_{missing}(N) &\propto p_c \cdot T_{ab} + p_c \cdot \sum_{n=2}^{N} (1-p_c)^{n-1} \cdot T_{ab}^{n} \\ &= p_c \cdot T_{ab} \cdot [\mathbb{I} + \sum_{n=2}^{N} (1-p_c)^{n-1} \cdot T_{ab}^{n-1}] \\ &= p_c \cdot T_{ab} \cdot [\mathbb{I} + \sum_{n=1}^{N} [(1-p_c) \cdot T_{ab}]^{n}] \\ &= p_c \cdot T_{ab} \cdot \sum_{n=0}^{N} [(1-p_c) \cdot T_{ab}]^{n} \end{aligned} \tag{30}$$

And after normalization

$$\begin{aligned} T_{missing}(N) &= [PC \cdot \sum_{n=0}^{N} [(1-p_c)]^{n}]^{-1} \cdot p_c \cdot T_{ab} \cdot \sum_{n=0}^{N} [(1-p_c) \cdot T_{ab}]^{n} \\ &= [PC \cdot \frac{1-(1-p_c)^{N+1}}{p_c}]^{-1} \cdot p_c \cdot T_{ab} \cdot \sum_{n=0}^{N} [(1-p_c) \cdot T_{ab}]^{n} \\ &= \frac{p_c}{1-(1-p_c)^{N+1}} \cdot T_{ab} \cdot \sum_{n=0}^{N} [(1-p_c) \cdot T_{ab}]^{n} \end{aligned} \tag{31}$$

Where N is the number of gaps in the sentence. The highest eigenvalue of the stochastic matrix T_{ab} is 1, so for $p_c \in [0,1) \rightarrow |p_c \cdot T_{ab}| < 1$ and we can write :

$$T_{missing}(N) = \frac{p_c}{1-(1-p_c)^{N+1}} \cdot T_{ab} \cdot [\mathbb{I} - (1-p_c) \cdot T_{ab}]^{-1} [\mathbb{I} - (1-p_c)^{N} T^{N}] \tag{32}$$

$$T_{missing}(N) \xrightarrow[N]{\infty} p_c \cdot T_{ab} \cdot [\mathbb{I} - (1-p_c) \cdot T_{ab}]^{-1} \tag{33}$$

As 33 shows, given any T and p_c, we can infer $T_{missing}$ the transition matrix given $OP(\cdot)$. Notice that we define this transformation as the "backward transformation" earlier.
For the "forward transformation", which is the transformation from ($T_{missing}$,p_c) to T :

$$T_{ab} = [\mathbb{I} \cdot p_c + (1-p_c)T_{missing}[\mathbb{I} - [(1-p_c) \cdot T_{ab}]^N]^{-1}]^{-1} \cdot T_{missing} \cdot [\mathbb{I} - [(1-p_c) \cdot T_{ab}]^N]^{-1}$$
$$\xrightarrow[N]{\infty} [\mathbb{I} \cdot p_c + (1-p_c) \cdot T_{missing}]^{-1} \cdot T_{missing} \tag{34}$$

From now on we will present the backward transformation as T^{-p_c} and the forward transformation as T^{p_c}.

Lemma F.1. *π the stationary distribution of T_{ab} is equal to π_m the stationary distribution of $T_{missing}$ for $N \to \infty$*

Proof. Given π the stationary distribution of T_{ab}, and from (**??**)($K = [PC \cdot \sum_{n=0}^{N}(1-p_c)^n]^{-1}$ normalization factor) :

$$\begin{aligned} \pi \cdot T_{missing} &= K \cdot p_c \cdot \sum_{n=0}^{N}(1-p_c)^n \cdot \pi \cdot T_{ab}^{n+1} \\ &= K \cdot p_c \cdot \sum_{n=0}^{N}[(1-p_c)^n \cdot \pi] \\ &= \pi \cdot K \cdot p_c \cdot \sum_{n=0}^{N}(1-p_c)^n = \pi \end{aligned} \tag{35}$$

□

In fact we can say that given T_m a stochastic matrix build as a polynomial of stochastic matrix T, $T_m = \sum_i a_i \cdot T^i$ if π is a stationary vector for T it is also a stationary vector for T_m.

F.0.1. MORPHISM

Lemma F.2. *$(T^{-Q_0})^{-Q}$ = $T^{-Q \cdot Q_0}$ and $(T^{Q_0})^{Q}$ = $T^{Q \cdot Q_0}$*

Proof.

$$\begin{aligned} &(T^{-Q_0})^{-Q} = p_q \cdot T^{-Q_0} \cdot [\mathbb{I} - (1-p_q) \cdot T^{-Q_0}]^{-1} \\ &= p_q \cdot P_{Q_0} \cdot T_{ab} \cdot [\mathbb{I} - (1-P_{Q_0}) \cdot T_{ab}]^{-1} \cdot [\mathbb{I} - (1-p_q) \cdot P_{Q_0} \cdot T_{ab} \cdot [\mathbb{I} - (1-P_{Q_0}) \cdot T_{ab}]^{-1}]^{-1} \\ &= p_q \cdot P_{Q_0} \cdot T_{ab} \cdot [[\mathbb{I} - (1-P_{Q_0}) \cdot T_{ab}] \cdot [\mathbb{I} - (1-p_q) \cdot P_{Q_0} \cdot T_{ab} \cdot [\mathbb{I} - (1-P_{Q_0}) \cdot T_{ab}]^{-1}]^{-1} \\ &= p_q \cdot P_{Q_0} \cdot T_{ab} \cdot [[\mathbb{I} - (1-P_{Q_0}) \cdot T_{ab}] - (1-p_q) \cdot P_{Q_0} \cdot T_{ab}]^{-1} \\ &= (p_q \cdot P_{Q_0}) \cdot T_{ab} \cdot [[\mathbb{I} - (1-(p_q \cdot P_{Q_0})) \cdot T_{ab}]^{-1} = T^{-Q \cdot Q_0} \end{aligned} \tag{36}$$

Following our resent prove :

$T^{-Q \cdot Q_0} = (T^{-Q_0})^{-Q}$

We will use the forward transformation twice on both sides:

$((T^{-Q \cdot Q_0})^{Q})^{Q_0} = (((T^{-Q_0})^{-Q})^{Q})^{Q_0}$

More, we claim that $(T^{p_c})^{-p_c} = T$ because the forward transformation derived directly from the backward transformation so $((T^{-Q \cdot Q_0})^{Q})^{Q_0} = T$

Finally:

$$\begin{aligned}((T^{-Q\cdot Q_0})^Q)^{Q_0} = T = (T^{-Q\cdot Q_0})^{Q\cdot Q_0}\\ (T^Q)^{Q_0} = (T)^{Q\cdot Q_0}\end{aligned} \tag{37}$$

□

F.0.2. FORWARD AND BACKWARD COMPOSITION

We will start with calculating the composition of the forward to the backward transformations. This process can be described as trying to reconstruct T from $T_{missing}$ based on the wrong assumption of Φ_Ψ when the real process was Φ_{p_q}

$$\begin{aligned}&(T^{-p_q})^{p_c} = [I\cdot p_c + (1-p_c)\cdot p_q\cdot T\cdot [I-(1-p_q)\cdot T]^{-1}]^{-1}\cdot p_q\cdot T\cdot [I-(1-p_q)\cdot T]^{-1}\\ &((T^{-p_q})^{p_c})^{-1} = [I-(1-p_q)\cdot T]\cdot\frac{1}{p_q}\cdot T^{-1}\cdot [I\cdot p_c+(1-p_c)\cdot p_q\cdot T\cdot [I-(1-p_q)\cdot T]^{-1}]\\ &=\frac{1}{p_q}\cdot[T^{-1}-(1-p_q)\cdot I]\cdot[I\cdot p_c+(1-p_c)\cdot p_q\cdot T\cdot[I-(1-p_q)\cdot T]^{-1}]\\ &=\frac{1}{p_q}\cdot[T^{-1}\cdot p_c+(1-p_c)\cdot p_q\cdot T^{-1}\cdot T\cdot[I-(1-p_q)\cdot T]^{-1}\\ &\quad -I\cdot p_c\cdot(1-p_q)-(1-p_c)\cdot(1-p_q)\cdot p_q\cdot T\cdot[I-(1-p_q)\cdot T]^{-1}]\\ &=\frac{p_c}{p_q}\cdot T^{-1}+[(1-p_c)-(1-p_c)\cdot(1-p_q)\cdot T]\cdot[I-(1-p_q)T]^{-1}-I\cdot(1-p_q)\cdot\frac{p_c}{p_q}\\ &=\frac{p_c}{p_q}\cdot T^{-1}+(1-p_c)[I-(1-p_q)T]\cdot[I-(1-p_q)T]^{-1}-I\cdot(1-p_q)\cdot\frac{p_c}{p_q}\\ &=\frac{p_c}{p_q}\cdot T^{-1}+I\cdot[(1-p_c)-(1-p_q)\cdot\frac{p_c}{p_q}]\\ &=\frac{p_c}{p_q}\cdot T^{-1}+I\cdot\frac{p_q-p_c}{p_q}\end{aligned} \tag{38}$$

$$\begin{aligned}(T^{-p_q})^{p_c} &= [\frac{p_c}{p_q}\cdot T^{-1}+I\cdot\frac{p_q-p_c}{p_q}]^{-1}\\ &= [\frac{p_c}{p_q}\cdot T^{-1}+I\cdot(1-\frac{p_c}{p_q})]^{-1}\end{aligned} \tag{39}$$

Now we will use lemma F.2 to calculate $(T^{p_c})^{-p_q}$ by showing the forward and backward transformations are commutative :

$$\begin{aligned}(T^{p_c})^{-p_q} = (((T^{-p_q})^{p_q})^{p_c})^{-p_q} =\\ (((T^{-p_q})^{p_c})^{p_q})^{-p_q} = (T^{-p_q})^{p_c} =\\ [\frac{p_c}{p_q}\cdot T^{-1}+I\cdot(1-\frac{p_c}{p_q})]^{-1}\end{aligned} \tag{40}$$

We can formulate a connection between T,p_q and T^{-p_q} from (40). Lets assume we observed T^{-p_q} and someone gave us the transition matrix T, how can we infer p_q ? We can always use iterative method to solve this problem , but another option is to use the derivative of $\frac{d((T^{-p_q})^{p_c})^{-1}}{dp_c}$ which can be calculated regardless of the unknown p_q (p_c,$p_c+\epsilon$ can be chose randomly):

$\frac{d(((T^{-p_q})^{p_c})^{-1})}{dp_c} = \frac{1}{p_q} \cdot [T^{-1} - I]$. And :

$$\begin{aligned} p_q &= (\frac{d(((T^{-p_q})^{p_c})^{-1})}{dp_c})^{-1} \cdot [T^{-1} - I] \\ T &= [p_q \cdot \frac{d(((T^{-p_q})^{p_c})^{-1})}{dp_c} + I]^{-1} \end{aligned} \tag{41}$$

Figure 10 presents the sensitivity of the backward transformation to the p_c used for the transformation. A Markov chain with 10 states was generated where the transition matrix T_r was chosen at random(uniform). 1500 trajectories of length 100 were sampled proceeding by Φ_Ψ to build the empirical $T_{missing}$. Then, (3) was used to reconstruct T_r.

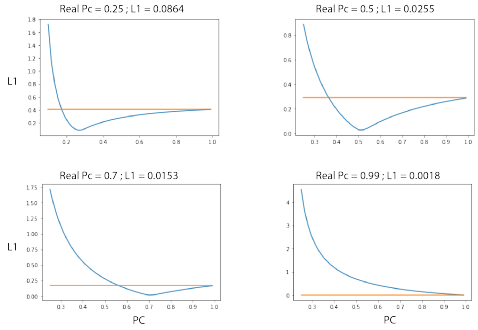


Figure 10. Backward transformation sensitivity to p_c. Each plot represents different ground truth p_c.Y-axis the L1 distance between T and T_r. X-axis the p_c for the backward transformation. Blue is the transformation results Orange the "naive" solution.

F.0.3. Robustness under iid misspecification

Lets now describe the simplest algorithm F = $Count(*)$ a function that count all the transitions in $\mathcal{O}$ and normalize to probability, as we know this is the maximum likelihood estimator for T with a Dirichlet prior. We define $T_P = Count(\Phi_\Psi(\mathcal{O}))$ and T_{M0} as the transition matrix of the original Markov chain
As we showed before :

$$T_P \propto p_c^2 \cdot T_{M_0} + p_c^2 \cdot \sum_{n=2}^{N} (1-p_c)^{n-1} \cdot T_{M_0}^n \tag{42}$$

And for $\hat{\Phi}$:

$$\begin{aligned} T_{\hat{P}} &\propto P(R_s) \cdot P(R_s|R_s) \cdot T_{M_0} + P(R_s) \cdot P(R_s|R_s) \cdot P(R_s|R_s) \cdot \sum_{n=2}^{N} (P(R_s|R_s)^{n-2} \cdot T_{M_0}^n \\ &= p_c \cdot (p_c + \epsilon) \cdot T_{M_0} + p_c \cdot (p_c - \epsilon) \cdot ((1-p_c) - \epsilon) \cdot \sum_{n=2}^{N} ((1-p_c) + \epsilon)^{n-2} \cdot T_{M_0}^n \\ &= T_{M_0}[p_c \cdot (p_c + \epsilon) \cdot I + p_c \cdot (p_c - \epsilon) \cdot \frac{1 - p_c - \epsilon}{1 - p_c + \epsilon} \cdot [\sum_{n=0}^{N} (1 - p_c + \epsilon)^n T_{M0}^n - I]] \\ &\xrightarrow[N]{\inf} T_{M_0} \cdot p_c \cdot (p_c + \epsilon) \cdot [I + \frac{(p_c - \epsilon)}{(p_c + \epsilon)} \cdot \frac{1 - p_c - \epsilon}{1 - p_c + \epsilon} \cdot [[I - (1 - p_c + \epsilon) \cdot T_{M0}]^{-1} - I]] \end{aligned} \tag{43}$$

While the normalize expression is

$$\begin{aligned} T_{M_0} \cdot (p_c+\epsilon) \cdot [I + \frac{(p_c-\epsilon)}{(p_c+\epsilon)} \cdot \frac{1-p_c-\epsilon}{1-p_c+\epsilon} \cdot [[I-(1-p_c+\epsilon)\cdot T_{M0}]^{-1} - I]] \\ T_{M_0} \cdot (p_c+\epsilon) \cdot [I + \frac{(p_c-p_c^2+\epsilon^2)-\epsilon}{(p_c-p_c^2+\epsilon^2)+\epsilon} \cdot [[I-(1-p_c+\epsilon)\cdot T_{M0}]^{-1} - I]] \end{aligned} \tag{44}$$

As we can see the difference lay in the ratio between the n-steps matrix and the original one. For $\epsilon \ll p_c$ we get :

$$\begin{aligned} T_{M_0} \cdot (p_c+\epsilon) \cdot [I + [[I-(1-p_c+\epsilon)\cdot T_{M0}]^{-1} - I]] \\ = T_{M_0} \cdot (p_c+\epsilon) \cdot [I-(1-p_c+\epsilon)\cdot T_{M0}]^{-1} \end{aligned} \tag{45}$$

Now lets investigate the robustness of our analytic reconstruction by evaluating the expected results given the new data distribution :

$$\begin{aligned} T_{reconstructed} &= [I \cdot p_c + (1-p_c)\cdot T_{\hat{P}}]^{-1} \cdot T_{\hat{P}} \equiv [I\cdot(1-p_c) + p_c \cdot T_{\hat{P}}^{-1}]^{-1} \\ &= [I\cdot(1-p_c) + \frac{p_c}{p_c+\epsilon} \cdot T_{M_0}^{-1} \cdot [I-(1-p_c+\epsilon)]\cdot T_{M_0}]]^{-1} \\ &= [I\cdot(1-p_c) + \frac{p_c}{p_c+\epsilon} \cdot [T_{M_0}^{-1} - (1-p_c+\epsilon)\cdot I]]^{-1} \\ &= [I\cdot\frac{\epsilon\cdot(1-2\cdot p_c)}{p_c+\epsilon} + \frac{p_c}{p_c+\epsilon}\cdot T_{M_0}^{-1}]^{-1} \end{aligned} \tag{46}$$

We can see that for $\epsilon \ll (1-p_c), p_c$ $T_{reconstruction} \approx T_{M_0}$.

G. Evaluations Models

In experiment 5b the values of p_c vary between sentences, hence, as σ increase, the *Naive* results present the relative effect of "fairly known" sentences in comparison to "badly known" ones. As all cases show, the benefit of "good" sentences is bigger than the disadvantage of "bad" ones. That is, non-constant p_c is an advantage to the *Naive* algorithm, but not for ours, resulting in diminishing advantage. Notice that on closer look, not all cases are affected equally, especially, in the "Part-Of-Speech"(POS) case the difference is more noticeable. We believe that the reasons for this are: 1) relatively short distance of the POS transition matrix from its stationary distribution. 2) similarity between each entry in the transition matrix to the stationary distribution. That is, not only T is similar to $T^d \xrightarrow[\infty]{d} \pi$ the fixed stationary distribution, it converges to π for smaller d's. So, the effect of increasing d (i.e increasing p_c) is diminishing faster. Figure 12 in the supplementary material compares the distance from stationarity for the different cases.

Figure 11 presents heat-maps of the transitions matrices for the four models used for evaluation.

Figure 12 presents the distance to stationary distribution for each model .

H. Standard Deviation (S.D) of the Experiments Results

For each figure in the paper, all the algorithms presented in the figure are evaluated on the same exact data and for the same random seed. The standard deviations(s.d) reported here are calculated as follow: for each figure (i.e "Known Specifications", "Wrong p_c", etc.), for each data model (i.e "Synthetic D=5", "Part Of Speech", etc.), we picked one representative algorithm (most relevant to the figure) and parameters (most challenging), and reported the s.d between different data and seeds.

All the s.d evaluations are presented in Table 2.

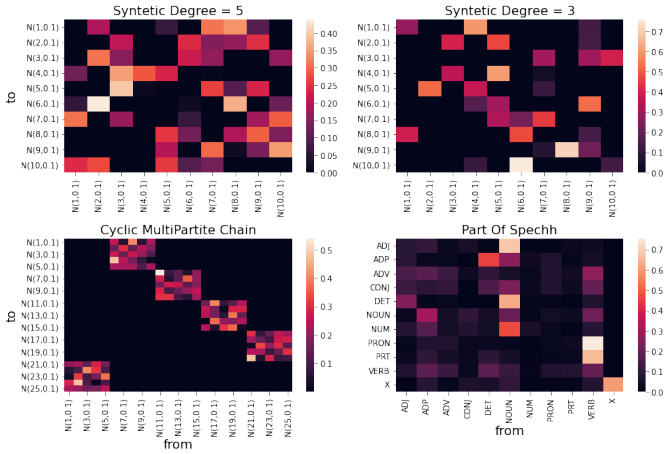


Figure 11. Heat maps of the transitions matrices of each model used in the paper.

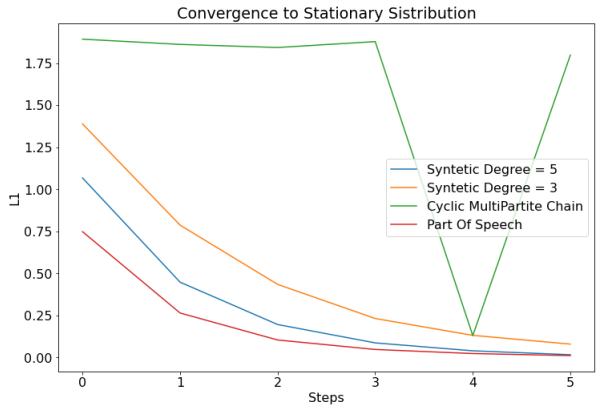


Figure 12. Distance to stationary distribution for each model used in the paper. X-label is the the number of steps for the d-step matrix T^d. Y-axis is the L1 distance between T^d and T

Table 2. S.D for paper experiments

Model	Experiment	Algorithm	Algorithm Input	Real Parameter	S.D
Syntetic D=5	Ignore Missing Observations	Naive	$\emptyset$	$\emptyset$	0.0604
Syntetic D=5	Known W	HMM	W	W	0.0048
Syntetic D=5	Known p_c	FOHMM	$p_c = 0.5$	$p_c = 0.5$	0.0048
Syntetic D=5	Known N	FOHMM	$N = 0.5$	$N = 80$	0.0231
Syntetic D=5	Wrong p_c	FOHMM	$p_c = 0.7$	$p_c = 0.5$	0.0117
Syntetic D=5	Non-constent p_c	FOHMM	$p_c = 0.5$	$p_c \sim N(.5, .2)$	0.0084
Syntetic D=5	Non-iid p_c	FOHMM	$p_c = 0.5$	$\epsilon = 0.15$	0.0276
Syntetic D=3	Ignore Missing Observations	Naive	$\emptyset$	$\emptyset$	0.0450
Syntetic D=3	Known W	HMM	W	W	0.0285
Syntetic D=3	Known p_c	FOHMM	$p_c = 0.5$	$p_c = 0.5$	0.0481
Syntetic D=3	Known N	FOHMM	$N = 0.5$	$N = 80$	0.0346
Syntetic D=3	Wrong p_c	FOHMM	$p_c = 0.7$	$p_c = 0.5$	0.0599
Syntetic D=3	Non-constent p_c	FOHMM	$p_c = 0.5$	$p_c \sim N(.5, .2)$	0.0509
Syntetic D=3	Non-iid p_c	FOHMM	$p_c = 0.5$	$\epsilon = 0.15$	0.1302
POS	Ignore Missing Observations	Naive	$\emptyset$	$\emptyset$	0.0423
POS	Known W	HMM	W	W	0.0162
POS	Known p_c	FOHMM	$p_c = 0.5$	$p_c = 0.5$	0.0079
POS	Known N	FOHMM	$N = 0.5$	$N = 80$	0.0107
POS	Wrong p_c	FOHMM	$p_c = 0.7$	$p_c = 0.5$	0.0089
POS	Non-constent p_c	FOHMM	$p_c = 0.5$	$p_c \sim$ N(.5,.2)	0.0091
POS	Non-iid p_c	FOHMM	$p_c = 0.5$	$\epsilon = 0.15$	0.0116
Cyclic MultiPartite Chain	Ignore Missing Observations	Naive	$\emptyset$	$\emptyset$	0.0094
Cyclic MultiPartite Chain	Known W	HMM	W	W	0.0083
Cyclic MultiPartite Chain	Known p_c	FOHMM	$p_c = 0.5$	$p_c = 0.5$	0.0091
Cyclic MultiPartite Chain	Known N	FOHMM	$N = 0.5$	$N = 80$	0.0144
Cyclic MultiPartite Chain	Wrong p_c	FOHMM	$p_c = 0.7$	$p_c = 0.5$	0.0067
Cyclic MultiPartite Chain	Non-constent p_c	FOHMM	$p_c = 0.5$	$p_c \sim$ N(.5,.2)	0.0072
Cyclic MultiPartite Chain	Non-iid p_c	FOHMM	$p_c = 0.5$	$\epsilon = 0.15$	0.0051