Latent variable models for hippocampal sequence analysis

by

Etienne Rudolph Ackermann

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree

Doctor of Philosophy

APPROVED, Thesis Committee:

Dr. Caleb Kemere, Chair
Associate Professor
Electrical and Computer Engineering
Rice University

Dr. Marina Vannucci
Noah Harding Professor and Chair
Department of Statistics
Rice University

Dr. Richard Baraniuk
Victor E. Cameron Professor
Electrical and Computer Engineering
Rice University

Dr. Daoyun Ji
Associate Professor
Molecular and Cellular Biology
Baylor College of Medicine

Houston, Texas
June 2019
Latent variable models for hippocampal sequence analysis

by

Etienne Rudolph Ackermann

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy

Approved, Thesis Committee:

Dr. Caleb Kemere, Chair
Associate Professor
Electrical and Computer Engineering
Rice University

Dr. Marina Vannucci
Noah Harding Professor and Chair
Department of Statistics
Rice University

Dr. Richard Baraniuk
Victor E. Cameron Professor
Electrical and Computer Engineering
Rice University

Dr. Daoyun Ji
Associate Professor
Molecular and Cellular Biology
Baylor College of Medicine

Houston, Texas
June 2019
PLACE cell activity of hippocampal pyramidal cells has been described as the cognitive substrate of spatial memory. Indeed, the activity of ensembles of neurons within the hippocampus is thought to enable memory formation, storage, recall, and even decision making. Replay is observed during hippocampal sharp-wave-ripple-associated population burst events (PBEs) and is critical for consolidation and recall-guided behaviors. Notably, these PBEs occur during times of inactivity, so that their representations cannot easily be matched with observable animal behavior.

In my thesis, I present an approach to uncover temporal structure within hippocampal output patterns during PBEs. More specifically, I use hidden Markov models (HMMs) to study PBEs observed in rats during exploration of both linear tracks and open fields, and I demonstrate that estimated models are consistent with a spatial map of the environment. Moreover, I demonstrate how the model can be used to identify hippocampal replay without recourse to the place code. These results suggest that downstream regions may rely on PBEs to provide a substrate for memory. Moreover, by forming models independent of animal behavior, I lay the groundwork for studies of non-spatial memory.

Next, I present a new model, the “clusterless” switching Poisson hidden Markov model, which extends my work on HMMs of PBEs to the case where we only have multiunit (unsorted) spikes. Indeed, spike sorting is challenging, time-consuming, often subjective (not reproducible), and throws away potentially valuable information from unsorted spikes, as well as our certainty about the cluster assignments. It has previously been shown that we can often do just as well, or in some cases even better, if we forego the spike sorting process altogether, and work directly with the unsorted data. Consequently, my clusterless HMM will enable us to combine the benefits of unsupervised learning for internally generated neural activity, with the benefits of clusterless approaches (more data leading to higher fidelity, especially at fine temporal scales, and additional probabilistic / soft information to exploit). I demonstrate the model’s ability to recover model parameters for simulated data, and show that it is able to learn a spatially-consistent representation of the environment from real experimental data.
This thesis is dedicated to
my parents, Thunis and Ilse,
and to Anke, Tiaan, and Klara;
to those whom I wish I could have spent more time with,
and to 黃美麟, my dearest.
I wish to extend my heartfelt gratitude to almost everyone, but in particular to my thesis advisor: Dr. Caleb Kemere—who has given me the freedom to pursue things that excited me, who has given me many opportunities and a lot of support throughout. I would also like to express my gratitude to the rest of my thesis committee: Drs. Baraniuk, Vannucci, and Ji, who have had to put up with my last-minute scheduling and strange timeline. I also want to thank Dr. Ji for hosting countless journal clubs and discussions (with pizza to boot!) that stimulated and enriched my understanding of learning and memory.

I am also indebted to the wonderful staff at Rice University, including the support staff in ECE and at OISS who have helped me time and again to get reimbursed, to get my paperwork filed in a timely manner, or simply with their friendly support and encouragement. Many thanks also to the Fulbright program, which is the primary reason why I started this PhD in the first place, and for all the incredible experiences they have afforded me, and the amazing people that I’ve met through their seminars and conferences and generosity.

I would like to express my gratitude to friends, both near and far, who have spent time with me at Rice, and who have supported me during my time here. I had incredibly fun times with RGCF and Bridges International in particular. To my labmates, thanks for your support and friendship over the years! Yes, even you, REL.

And finally my deepest appreciation goes out to my entire family, from all around the world, who have cheered me on, hosted me, and supported me through good times and bad. You were my strength when I felt ready to give up. Thank you all.

Don’t miss out! Become (unofficially) acknowledged!

If I have failed to mention you explicitly, please amend this thesis by printing and completing this page, and consider yourself acknowledged.

I, Etienne Rudolph Ackermann, hereby—somewhat reluctantly—acknowledge ________ for the (☐ insignificant ☐ small ☐ pivotal ☐ weird ☐ epic ☐ other) contribution that (s)he has made.
# Table of Contents

Table of Contents  

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Overview</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Organization of this thesis</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Contributions of this thesis</td>
<td>4</td>
</tr>
<tr>
<td><strong>2 Background information and literature review</strong></td>
<td>5</td>
</tr>
<tr>
<td>2.1 Representing space in the hippocampus</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Latent variable models of neural activity</td>
<td>11</td>
</tr>
<tr>
<td><strong>3 Hidden Markov models for exploratory data analysis</strong></td>
<td>13</td>
</tr>
<tr>
<td>3.1 Uncovering temporal structure in hippocampal output patterns</td>
<td>14</td>
</tr>
<tr>
<td>3.1.1 Overview</td>
<td>14</td>
</tr>
<tr>
<td>3.1.2 Learning hidden Markov models from PBE data</td>
<td>15</td>
</tr>
<tr>
<td>3.1.3 Hidden Markov models from PBEs capture the place code</td>
<td>23</td>
</tr>
<tr>
<td>3.1.4 HMM-congruent PBEs capture sequence replay</td>
<td>26</td>
</tr>
<tr>
<td>3.1.5 HMMs for unconstrained environments</td>
<td>32</td>
</tr>
<tr>
<td>3.2 Remote replay detection using hidden Markov models</td>
<td>35</td>
</tr>
<tr>
<td>3.2.1 Awake replay of remote experiences</td>
<td>35</td>
</tr>
<tr>
<td>3.2.2 Hidden Markov models for remote replay detection</td>
<td>38</td>
</tr>
<tr>
<td>3.2.3 Non-spatial and unobserved remote contexts</td>
<td>43</td>
</tr>
<tr>
<td>3.3 Discussion</td>
<td>46</td>
</tr>
<tr>
<td>3.3.1 Graded, non-binary replay detection</td>
<td>47</td>
</tr>
<tr>
<td>3.3.2 Learning, model congruence and replay quality</td>
<td>48</td>
</tr>
<tr>
<td>3.3.3 Application to Extra-spatial Behaviors</td>
<td>48</td>
</tr>
<tr>
<td>3.3.4 Conclusions</td>
<td>49</td>
</tr>
<tr>
<td><strong>4 Clusterless hidden Markov models of neural activity</strong></td>
<td>51</td>
</tr>
<tr>
<td>4.1 Motivation</td>
<td>51</td>
</tr>
<tr>
<td>4.2 Model specification</td>
<td>53</td>
</tr>
<tr>
<td>4.2.1 Notation and preliminaries</td>
<td>53</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.2.2 Sampling approach to evaluate $P(y_t</td>
<td>S_t)$</td>
</tr>
<tr>
<td>4.2.3 Updating the state-dependent firing rates, $r_t^{(j)}$</td>
<td>58</td>
</tr>
<tr>
<td>4.3 Clusterless HMM Results</td>
<td>59</td>
</tr>
<tr>
<td>4.3.1 Simulated data</td>
<td>59</td>
</tr>
<tr>
<td>4.3.2 Experimental data</td>
<td>66</td>
</tr>
<tr>
<td>4.4 Related work</td>
<td>68</td>
</tr>
<tr>
<td>4.5 Discussion</td>
<td>70</td>
</tr>
<tr>
<td>5 Conclusion and future directions</td>
<td>71</td>
</tr>
<tr>
<td>Bibliography</td>
<td>73</td>
</tr>
<tr>
<td>List of Abbreviations</td>
<td>84</td>
</tr>
</tbody>
</table>
# List of Figures

3.1 Data preprocessing workflow for switching Poisson hidden Markov models of neural activity. ........................................... 18
3.2 A hidden Markov model of ensemble activity during PBEs. ........ 19
3.3 Models of PBE activity are sparse .................................... 22
3.4 Latent states capture positional code ................................. 24
3.5 Decoding accuracy for different numbers of states ................. 25
3.6 Sequence replay detection using HMMs ............................. 29
3.7 Bayesian replay detection vs HMM congruence ...................... 30
3.8 Modeling PBEs in open field. ....................................... 33
3.9 Remote replay task illustration .................................... 36
3.10 Examples of linearized replay detection using regression analysis . . 37
3.11 IsPFs of remote context E1 ....................................... 40
3.12 Examples of remote replay candidates scored with the HMM ...... 41
3.13 Example of remote replay candidate correctly rejected by HMM ... 42
3.14 Stable model quality after removing local spatially-consistent events suggests remote or non-spatial replay ........................ 44
3.15 Locally-congruent but Bayesian non-significant PBEs corresponding to remote replay events .................................... 45

4.1 Probabilistic graphical model for the clusterless HMM ............. 57
4.2 Simulated mark data generation diagram .......................... 60
4.3 Clusterless HMM parameter estimation and decoding on a small simulated dataset ........................................... 61
4.4 Clusterless HMM parameter estimation and decoding on a medium-size simulated dataset ................................. 62
4.5 Over specifying number of hidden neurons simply partition the feature space more finely ................................. 63
4.6 Over-specifying the number of hidden neurons has almost no effect on parameter estimation ................................. 64
4.7 Over-specifying the number of hidden states has almost no effect on decoding accuracy ................................... 65
LIST OF FIGURES

4.8 Feature space partitioning of a single probe from the linear track data 66
4.9 Clusterless HMM decoding of linear track data .......................... 67
LIST OF TABLES

3.1 Regression vs HMM-based remote replay detection .................................. 40
3.2 Pooled comparison of regression vs HMM detection of remote replay ....... 43
Chapter 1

Introduction

“There is no scientific study more vital to man than the study of his own brain. Our entire view of the universe depends on it.”

Francis Crick

Spatial neuronal firing patterns are often preserved and repeated in a time-compressed manner, but it remains challenging to identify and quantitatively assess these replay events from extracellularly recorded neural data. These replay events are believed to play a critical role in memory consolidation, and their identification and analysis will help neuroscientists to better understand memory formation, consolidation, and cognition. Two approaches to identify replay events in the hippocampus are commonly used, namely (i) a template matching approach (which fails when small subsets of a neuronal population participate in a replay event), and (ii) a more advanced Bayesian approach, in which the sequence of spikes is decoded to a sequence of positions (or other behavioral correlate) in an environment as observed during a spatial navigation task. The quality of a replay event is then determined by comparing this series of decoded positions to ordered trajectories through the environment, as well as to a large set of randomly shuffled decoded positions. However, this decoding approach is difficult to apply to regions of the brain where a clear spatial context is not apparent, and even when the context is clearly spatial, many of the candidate replay events simply don’t have enough (sorted) spikes to decode anything useful at such small time scales. Consequently, we set out to develop approaches to deal
with both of these challenges; in particular, we set out to develop (i) an unsupervised approach for replay analysis and detection so that we can analyze candidate events in the absence of animal behavior, and (ii) a clusterless extension of (i) which will allow us to include information from unsorted spikes and spike waveform features to improve our decoding ability at small time scales.

1.1 Overview

The activity of ensembles of neurons within the hippocampus is thought to enable memory formation, storage, recall, and potentially decision making. In rodents, hippocampal “place cells” are known to encode an animal’s location in its environment as it explores (O’Keefe and Dostrovsky, 1971). Hence, populations of these neurons fire in temporally-ordered sequences corresponding to the spatiotemporal trajectories the animals traverse. Of particular interest to us are hippocampal replay events in which neurons recapitulate their spatially-ordered sequences during periods of quiescence or sleep (and often associated with brief, 150–250 Hz Sharp Wave Ripple (SWR) oscillations in the hippocampus).

Most approaches to replay detection rely on the estimation of behavioral templates during active behavior, followed by comparisons of the SWR-associated replay candidate events to the learned behavioral templates. Such an approach is critically dependent on (i) the availability of behavioral data, as well as (ii) the associated task complexity.

The requirement to have access to the behavioral correlates have caused studies of hippocampal replay to be limited to those where the behavioral correlates are well understood and easily observable (most notably that of position); it is much more difficult to identify sequences of non-spatial memories e.g., sequences of odor cues. In addition, template matching approaches quickly become prohibitive even for relatively simple tasks, and therefore more powerful, generalizable approaches have to be considered (see e.g. Pfeiffer and Foster (2013) for an alternative approach to template matching, where heuristic rules such as a maximum jump distance from one frame to the next, and a minimum end-to-end distance, were used in an open field).

Hidden Markov models (HMMs) are well suited to model this sort of sequential activity—due in part to the Markovian nature of spatial locomotion (e.g., our position at time $t$ is constrained by our position at time $t - 1$), but perhaps more importantly
is the fact that, by definition, there is no observable animal behavior during candidate replay events. Indeed, HMMs have found several uses in neuroscience (see for example Kemere et al., 2008a; Florian et al., 2011), and have also been used to uncover hippocampal population codes during awake (Linderman et al., 2016), as well as during sleep-associated activity (Chen et al., 2016).

In this thesis, we focus primarily on the HMM as a tool for exploratory data analysis, and we develop approaches to identify replay-like (so-called model congruent) events, to identify remote replay with or without access to the remote context’s neural activity (and always unsupervised in the sense that we do not pass behavioral data to the model), as well as ways to interpret the models beyond sequences in the state space.

We also develop a novel clusterless extension of the HMM, with which we can learn latent representations of spike data directly from waveform features, without the need for spike sorting. Such an approach holds the promise to analyze short-time-duration events (such as replay events, or theta sequences) with higher fidelity, due in part to the additional information from unsorted spikes, as well as the probabilistic information characteristic of clusterless approaches.

Finally, we also present ongoing work on an open-source, publicly-available Python framework for electrophysiology data analysis, that has helped me a great deal in completing all the analyses presented here.

1.2 Organization of this thesis

The most important and relevant literature is presented in Chapter 2, followed by our work on modeling hippocampal Population Burst Events (PBEs) using HMMs in Chapter 3. We present an approach to build models directly from internally generated PBEs in section 3.1, and show that such models capture the underlying place code (section 3.1.3), and how they can be used to detect replay-like sequences (section 3.1.4). We also demonstrate how we can use our HMMs to detect instances of remote replay (section 3.2.2, as well as potentially non-spatial memories (section 3.2.3).

In Chapter 4 we present a new model, the clusterless hidden Markov model, as an extension of current clusterless approaches to the unsupervised case, or alternatively, as an extension of our unsupervised HMM approach to the clusterless setting in which we do not need to do spike sorting, and instead fit the model directly on the spike
waveform features. We detail the new model in section 4.2, and present results on both simulated data (section 4.3.1) as well as real experimental data (section 4.3.2).

1.3 Contributions of this thesis

In this thesis, I present

| a new analytical framework for studying hippocampal ensemble activity using HMMs. |

This framework allows us to build models directly from PBEs, without strong assumptions on what the PBEs might encode. This is in contrast to existing approaches in which the model—estimated place fields for the ensemble—is formed using the theta-associated place cell activity, after which PBEs are interpreted with reference to the place cell model. Moreover, I present

| an approach to use these PBE models to detect replay-like events, completely independent of animal behavior, |

as well as approaches to use our HMM framework to

| detect remote- and potentially non-spatial replay events. |

To the best of our knowledge, this is the first time models have been built directly from PBEs, and that replay identification has been performed entirely independent of animal behavior (even Box et al. (2014), who used an HMM to detect replay, did so with the incorporation of animal behavior).

Furthermore, I present

| a new clusterless (spike sorting free) extension to the HMM, |

which is the first unsupervised clusterless (decoding) approach, as well as the first attempt to “bin” a marked point process into observation windows.

Some of the work presented here has already been published in Ackermann and Kemere (2016); Ackermann et al. (2017); Maboudi et al. (2018), and additional publications are being prepared for my work on the clusterless HMM, as well as the Python framework.
Chapter 2

BACKGROUND INFORMATION AND LITERATURE REVIEW

“One of the difficulties in understanding the brain is that it is like nothing so much as a lump of porridge.”

Richard Gregory

ONE of the most fascinating and exciting discoveries in neuroscience within the last 50 years is arguably that of so-called place cells by O’Keefe and Dostrovsky (1971). Place cells are pyramidal neurons in the hippocampus which become active only in particular locations in an environment, thereby helping to form a cognitive map (O’Keefe and Nadel, 1978) which has been shown support spatial navigation. More specifically, each place cell has an associated region (referred to as its place field, or sometimes its receptive field) where it exhibits increased firing activity. Inside its place field, a place cell could have a maximum firing rate of about 25 Hz or more, while having a firing rate of less 0.1 Hz outside of its field.

Interestingly, there is no apparent topology to the pattern of place fields within the hippocampus. That is, adjacent place cells could have very distant place fields in one environment, and neighboring or overlapping place fields in the next (O’Keefe et al., 1998). This lack of organization is in contrast to some other brain regions—such as the visual cortex—where a fairly robust topography can be observed (see e.g., Miller et al., 1989). Nevertheless, place cells often exhibit remarkable stability, with many of
the place cells being recruited to represent the same place fields as before when an animal revisits a previously-explored environment (Save et al., 2000).

Since the discovery of place cells, many functionally-related cells have been discovered, including grid cells, head direction cells, boundary cells, and more. All of these different types of cells are interesting and important on their own, but they are perhaps even more interesting when studied together with place cells, where we are slowly approaching a unified understanding of spatial representation and planning in the brain (see Buzsáki and Moser, 2013, for a recent review). However, there are still many unanswered questions about how this spatial representation system really works.

One of the unanswered questions surrounds the functional significance of a phenomenon called hippocampal replay (replay for short), where spatiotemporal neuronal firing patterns are preserved, and (frequently) repeated at a later stage in a time-compressed manner. These repeated neuronal firing sequences have been observed in both sleeping and awake animals, and is thought to support (among other things) memory consolidation and cognition (Carr et al., 2011).

Unfortunately, it remains challenging to identify and quantitatively assess these replay events from extracellularly recorded neural data. Studying replay is inherently a challenging task: not only is it still relatively poorly understood (and its functional role and significance is still being explored), but the optimal way in which to detect replay events remains unclear.

More specifically, unlike with a well-defined stimulus-response type experiment where we can quantitatively assess decoding performance (e.g., how well can we predict an animal’s location—the stimulus—given its neural activity, i.e. the response), we do not have direct access to the “stimulus” associated with replay events. We are therefore usually forced to make fairly strong assumptions about what the replay events represent (such as mentally navigating through an environment), before even attempting to detect or interpret any such events. We try to break away from these constraints in this thesis, by developing a fully unsupervised approach to replay detection.
2.1 Representing space in the hippocampus

As mentioned previously, an animal’s location can be encoded by ensembles of place cells, each of which has an associated place field where the corresponding place cell is active.

**Place cells are randomly recruited** Populations of pyramidal cells (in particular place cells in hippocampal subfield CA1) have been shown to represent an animal’s location in an environment with remarkable accuracy. However, not all of the pyramidal cells are necessarily place cells. Indeed, one pyramidal cell might be a place cell in one environment, but not in another, so that place cells are “recruited”—seemingly at random—as they are needed. In an innovative study utilizing the immediate-early gene Arc to label active CA1 neurons, it has been shown that approximately 45% of pyramidal cells had place fields in any given environment (Guzowski et al., 1999), which is consistent with previously reported estimates (Gothard et al., 1996; Wilson and McNaughton, 1993).

**Decoding reveals non-local representations** Place cells sometimes exhibit reactivation outside of their place fields. The highly accurate hippocampal neural representation of an animal’s location in an environment then allows us to determine if and when non-local information is being represented in the hippocampus (Gupta, 2011), for example when an animal is “mentally navigating” through an environment. Such non-local sequences might be related to planning, episodic memory, or memory consolidation.

**Space is represented at three time-scales** Of course, the representation of space is not as simple as it might at first appear. More specifically, the hippocampus seems to represents space on three different time scales Gupta (2011), namely (i) the *behavioral time-scale* (on the order of a few seconds), where place cells are activated in sequence as an animal moves through its environment, (ii) the *theta time-scale* ($\approx 125$ ms), where phase-locked sequences of spikes within one cycle of theta oscillation ($\approx 8$ Hz) encode a short path that typically begins slightly behind the animal and ends slightly ahead of the animal, and (iii) the *replay time-scale* (lasting around 60–250 ms), where sequences of neural activity are observed during SWR complexes in restful behavior, including both sleep and awake states.
Increased neural activity during inactive states led to the discovery of replay

During awake behavior there are two main network states in the hippocampus (Buzsáki et al., 1983; Gupta, 2011), namely an active state, where the Local Field Potential (LFP) is characterized by having high power in the theta frequency band (4–12 Hz), and the inactive (or perhaps more accurately, the inattentive) state, where the power in the theta frequency band is much lower, and instead the LFP is characterized by large amplitude irregular activity.

The active state occurs when an animal is actively exploring an environment, or when they are stationary but attentive and engaged in a task (Gupta, 2011). In contrast, the inactive state occurs when animals are awake and eating, grooming, or resting, as well as during Slow Wave Sleep (SWS). The inactive state is generally characterized by reduced neural activity (as compared to the active state), but during brief (40–120 ms) high-frequency oscillations (150–250 Hz) known as SWR complexes, periods of increased neural activity have been observed (O’Keefe and Nadel, 1978; Buzsáki, 1989).

It has previously been suggested that encoding and retrieval of experiences may occur during the active (or theta) state (Skaggs et al., 1996; Tsodyks et al., 1996), but also that the high-frequency neural activity during SWRs in the inactive state may be well suited for memory transfer and consolidation from the hippocampus to the cortex (Buzsáki, 1989; Sutherland and McNaughton, 2000). This suggestion of memory consolidation during inactive states led to the following hypothesis:

“If neural activity during inattentive awake states and during sleep reflects a consolidation process involving the transfer of information from hippocampus to cortex, the neural activity would likely represent information from the animal’s recent experiences.” (Gupta, 2011)

This hypothesis was tested by Wilson and McNaughton (1993), where the activity of pairs of place cells were analyzed during active behavior and subsequent SWS. In particular, they identified pairs of neurons that fired together (within 100 ms of each other), and they found that pairs that fired together during active behavior preferentially fired together during SWRs in SWS, and that moreover, pairs that did not fire together during the active state, were similarly less likely to fire together.
during SWRs in the inactive (or SWS) state. This early work was quickly extended to show that the neuron pairs were not simply correlated, but that the order of firing was also preserved (Skaggs and McNaughton, 1996), and moreover, that the ordered firing was not limited to pairs of neurons, but that it extended to sequences of neurons (Nádasdy et al., 1999; Lee and Wilson, 2002). More specifically, place cell sequences have been shown to reactivate in a consistent sequential order both during Rapid Eye Movement (REM) sleep (Louie and Wilson, 2001) and during SWS (Lee and Wilson, 2002). Since the order during sleep frequently matched the order in which the corresponding cells were active during awake behavior, this phenomenon came to be known as hippocampal replay (Buhry et al., 2011).

**Reactivation of hippocampal activity** Reactivation of neural activity in the hippocampus has now been studied for over twenty five years, and although the very first study was published as early as 1989 (see Pavlides and Winson, 1989), the above-mentioned work by Wilson and McNaughton (1993) in 1993 is often considered the seminal contribution to the study of hippocampal reactivation.

Unfortunately, a lot still remains unknown about hippocampal reactivation, and replay in particular. Nevertheless, there have been many interesting developments in hippocampal replay over the last number of years, hinting at our limited (but increasing) understanding of replay and its functional significance. Below I summarize some of the most important findings about hippocampal replay.

**Replay events frequently occur during SWRs** It has been shown that reactivation largely co-occurs with hippocampal SWRs, but it remains unclear whether all SWRs are accompanied by replay events, and vice versa (Buhry et al., 2011).

**Replay can occur during awake or sleep states** Replay was first observed during sleep (and SWS in particular), but it has since been observed in the awake state too (see e.g., Nádasdy et al., 1999; Louie and Wilson, 2001; Carr et al., 2011).

**Replay can occur in a forward or reverse direction** Similarly, replay was originally thought to occur only in a forward direction. That is, in a temporal order which corresponds to the temporal order during the awake behavior. However, during some SWRs in the awake state, replay was observed in the reverse order (Foster and Wilson, 2006; Diba and Buzsáki, 2007).
Replay can represent remote locations It was also shown that awake replay events can represent remote locations (Karlsson and Frank, 2009) unlike the initial understanding that awake replay depends on sensory cues in the local environment.

Replay can span multiple SWRs Furthermore, it has been shown that replay events can span multiple SWRs to represent extended experiences (Davidson et al., 2009).

Replay is not simply a function of past experience Finally, one of the most recent and significant contributions to our understanding of replay was the work by Gupta et al. (2010), where it was shown that replay is not simply a function of past experience, but that trajectories which have never been experienced before (such as shortcuts through the environment) also appear frequently during replay, and they also showed that replay might serve a maintenance purpose, keeping less-often observed but relevant experiences from being lost. The existence of shortcut paths during replay seems to suggest that replay could play a role in learning, planning, and the maintenance of the cognitive map (Gupta, 2011).

Replay plays an important role in learning and memory Irrespective of the exact functional role or significance of hippocampal replay, it should already be clear that something interesting is happening during replay events, and that it could very likely play a role in memory consolidation, learning, and more.

Indeed, studies in which the SWRs were disrupted (see e.g., Jadhav et al., 2012; Ego-Stengel and Wilson, 2010; Girardeau et al., 2009) have demonstrated that the performance on hippocampal-dependent tasks were negatively affected by these disruptions, providing further support for the notion that SWRs and the associated neural activity (i.e., replay) play an important role in learning and memory.

Even though much of the previously mentioned work referred to SWRs, in this thesis I will use the closely-related PBEs, which is detected slightly differently\(^1\), but which can almost be thought of as synonymous with SWRs.

\(^1\)SWRs are detected by filtering the LFP to the appropriate band, and finding epochs of high power in the ripple band. We can therefore detect SWRs even in the absence of any recorded spikes. In contrast, PBEs are detected from multiunit (spike) activity, by finding epochs of high firing activity during behavioral inactivity.
2.2 Latent variable models of neural activity

Latent variable models span a range of uses, including dimensionality reduction (see e.g., Low et al., 2018; Williams et al., 2018), to neural decoding (Kemere et al., 2008a). See also Paninski and Cunningham (2018) and the references contained therein for an overview. However, we will primarily focus on the HMM here, which is an instance of a latent variable model.

Hidden Markov models have been very fruitfully used to understand sequentially structured data in a variety of contexts. A hidden Markov model captures information about data in two ways. First, it clusters observations into groups (“states”) with shared patterns. This is equivalent to reducing the dimension of the ensemble observations into a discretized latent space or manifold. Second, it models the dynamics of state transitions. This model is Markovian because it is assumed that the probability to transition to the next state only depends on the current state. Critically, these operations of clustering and sequence modeling are jointly optimized, allowing the structure of ensemble firing corresponding to each of the final states to combine information over many observations.

HMMs have found widespread use in neuroscience, too, ranging from studies of behavior, to neuron assemblies, and to individual ion channels (see Florian et al., 2011, for a good review). The observed variable could include an animal’s behavior, including a decision or a motor output, and in other cases the observed variable is the neural activity itself. HMMs have frequently been used to decode neural activity (see e.g., Pawelzik et al., 1993, who used an HMM to distinguish between periods of oscillatory versus stochastic firing in the cat’s visual cortex). Similarly, Gat and Tishby (1993) modeled monkey cortical activity as a multivariate time-dependent Poisson process (also referred to as a switching Poisson process), and Kemere et al. (2008a) used several HMMs (including a left-to-right HMM) to detect neural state transitions (baseline to plan to peri-movement) and trajectories towards goals for motor cortical prostheses in monkey.

HMMs have also found use in preprocessing steps in neuroscience, or as tools as part of some other analysis. For example, HMMs have been used for spike alignment to determine similarity between spike trains (Victor and Purpura, 1997; Florian et al., 2011), to remove artifacts (Dombeck et al., 2007), to align birdsong (Kogan and
Margoliash, 1998), and even for spike sorting (Herbst et al., 2008).

An interesting application area for the HMM is in the study of hippocampal population activity. Chen and Wilson (2013); Chen et al. (2013, 2014a, 2016) used a HMM to infer the spatial topology of environments, and to build unsupervised models of neural population codes. These efforts have subsequently been extended to nonparametric Bayesian extensions (including semi-Markov variants) by Linderman et al. (2016).

In this thesis, we will use the switching Poisson HMM framework as used by Kemere et al. (2008a); Chen and Wilson (2013) and others, but we will fit our models on internally-generated hippocampal PBEs to study replay-like sequences. Furthermore, our clusterless HMM takes elements from the marked point process filter framework presented by Deng et al. (2015), and combines it with the switching Poisson framework mentioned above.
Chapter 3

Hidden Markov models for exploratory data analysis

“Life is all memory, except for the one present moment that goes by you so quickly you hardly catch it going.”

Tennessee Williams

HMMs are most commonly used in a predictive / smoothing mode (see e.g., Kemere et al. (2008b); Ahmadian et al. (2011)). Indeed, Rabiner (1989) lists three common problems in HMMs namely

1. “Given an observation sequence, how do we efficiently compute the probability of the observation sequence, given the model?”,

2. “Given the observation sequence, how do we choose an optimal corresponding state sequence?”, and

3. “How do we estimate the model parameters?”

We are certainly interested in all of these questions, but we want to utilize the HMM as an exploratory tool as well. That is, we want to answer questions like “Is this seemingly-random dataset structured in some meaningful way? If so, how structured is it?”, or “Can we identify patterns of activity that seem to repeat more often than we would expect by chance?” and so on. As a result, we are often less interested in our model’s decoding ability, but investigating how well we are able to decode nevertheless
Hidden Markov models for exploratory data analysis

Chapter 3

gives us great insight into our model, as well as confidence that the model has learned something meaningful from the data.

In this chapter, we will look at two examples of using HMMs in an exploratory mode for sequential neural data analysis. In section 3.1 we will use HMMs to learn models of neural activity directly from PBEs—short bursts of internally generated neural activity—which will allow us to identify repeating patterns reminiscent of hippocampal replay, and in section 3.2 we will use HMMs to identify instances of remote replay in a nonlinearized w-maze, as well as some preliminary evidence that we can identify remote-replay-like events even in the absence of any knowledge of, or data from, the remote contexts.

3.1 Uncovering temporal structure in hippocampal output patterns

The work presented in this section has been published in Maboudi et al. (2018), and the interested reader is referred to the article for more information or a slightly different perspective. Some passages have been quoted verbatim, or with minimal modification.

3.1.1 Overview

In this section we develop an approach to answer a simple question:

Is there (significant) sequential structure in internally generated hippocampal neural activity during offline states?

We already suspect that the answer to this question is “yes”; however, this question has never been answered directly. Instead, previous studies have built templates of behavior, and subsequently matched a small number of internally generated events to those templates (Tatsumo et al., 2006). The implicit assumption is therefore that internally generated events resemble / recapitulate neural activity observed during behavior—and more subtly, that many of the “non-significant” events are either noise, failed sequences, or perhaps representations of other experiences. In either case, we wanted to answer the question more directly, without having to assume that internally generated events recapitulate those observed during behavior.
Place cell activity of hippocampal pyramidal cells has been described as the cognitive substrate of spatial memory. Replay is observed during hippocampal sharp-wave-ripple-associated PBEs and is critical for consolidation and recall-guided behaviors. PBE activity has historically been analyzed as a phenomenon subordinate to the place code.

In this section, we use HMMs to study PBEs observed in rats during exploration of both linear mazes and open fields. We demonstrate that estimated models are consistent with a spatial map of the environment, and that those models can even decode animals’ positions during behavior (section 3.1.3). Moreover, we develop and demonstrate an approach to identify hippocampal replay without recourse to the place code, using only PBE model congruence (section 3.1.4). These results suggest that downstream regions may rely on PBEs to provide a substrate for memory. Additionally, by forming models independent of animal behavior, we lay the groundwork for studies of non-spatial memory (see section 3.2.3).

3.1.2 Learning hidden Markov models from PBE data

Here we use the so-called switching Poisson hidden Markov model to describe population level neural activity (see e.g., Kemere et al., 2008a), and we loosely follow the approach and notation presented by Ackermann et al. (2017).

3.1.2.1 Awake population burst events

We analyzed the activity of large numbers of individual neurons in areas CA1 and CA3 of the dorsal hippocampus as rats navigated linear mazes for water reward (linear track: \( n = 3 \) rats, \( m = 18 \) sessions (previously used by Diba and Buzsáki, 2007), as well as from area CA1 as rats explored a \( 2 \) m \( \times \) 2 m open field for liquid reward (data was recorded by, and reported on by Pfeiffer and Foster, 2013, 2015). Using pooled multiunit activity, we detected PBEs during which many neurons were simultaneously active. The majority of these events occurred when animals paused running (speed \(< 5 \) cm/s, corresponding to \( 54.0\% \pm 20.1\% \) sd of events; linear track) to obtain reward, groom, or survey their surroundings (Buzsáki et al., 1983), and were accompanied by SWR, characterized by a burst of oscillatory activity in the 150–250 Hz band of the CA1 LFP. Because we are interested in understanding internally generated activity during PBEs, we included only these periods without active behavior, ensuring that theta sequences would not bias our results. While
we identified active behavior using a speed criterion, our collaborators found similar
results when they used a theta-state detection approach instead (not shown). We did
not add any other restrictions on behavior, LFPs, or the participation of place cells.
We found that inactive PBEs occupied an average of 1.8% of the periods during which
animals were on the linear track (16.9 ± 15.1 s of 832.6 ± 390.5 s). In comparison,
classical Bayesian approaches to understand PBE activity require the 34.8% of time
animals are running (speed > 10 cm/s) on the track (254.4 ± 106.6 s of 832.6 ± 390.5 s)
to build models of place fields.

Activity during PBEs is widely understood to be internally-generated in the
hippocampal-entorhinal formation, and likely to affect neuronal firing in downstream
regions (Buzsáki, 1989; Chrobak and Buzsáki, 1996; Logothetis et al., 2012; Yamamoto
and Tonegawa, 2017). Given the prevalence of PBEs during an animal’s early ex-
perience, we hypothesized that despite their short durations and noisy nature, the
neural activity during these events might nevertheless be sufficient to train an HMM of
sequential patterns, and that this model would capture the relevant spatial information
encoded in the hippocampus, independent of exploration itself.

Note that we intend to build models using only seconds’ worth of data---and noisy, internally generated data at that---as opposed to the minutes’ worth of robust behavioral data used in classical Bayesian decoding or template matching approaches.

3.1.2.2 Model specification

HMMs capture information about data in two ways. First, it clusters observations
into groups (“states”) with shared patterns. In our case, this corresponds to finding
time bins in which the same sets of neurons are co-active. This is equivalent to reducing
the dimension of the ensemble observations into a discretized latent space or manifold.
Second, it models the dynamics of state transitions. This model is first order Markov
because it is assumed that the probability to transition to the next state only depends
on the current state. Critically, these operations of clustering and sequence modeling
are jointly optimized, allowing the structure of ensemble firing corresponding to each of
the final states to combine information over many observations. Given the role of the
hippocampus in memory, in our HMMs, the unobserved latent variable presumably
corresponds to the temporal evolution of a memory trace that is represented by
co-active ensembles of CA1 and CA3 neurons. The full model will correspond to the structure which connects all the memory traces activated during PBEs.

Let \( y_t \) denote the observation at time \( t \), where \( y_t \in \mathbb{Z}_{\geq 0}^N \) is a vector of spike counts for \( N \) hippocampal pyramidal cells/neurons. It is assumed that the observations are sampled at discrete, equally-spaced time intervals, so that \( t \) can be an integer-valued index, with some associated \( \Delta t \).

We further assume that the hidden state space is discrete, and that it can take on one of \( Z \) possible states: \( S_t \in \{1, \ldots, Z\} \).

To define a probability distribution over sequences of observations, we then need to specify a probability distribution over the initial state \( P(S_1) \), with \( \pi_i \equiv \Pr(S_1 = i) \), the \( Z \times Z \) state transition probability matrix, \( A \), with \( A_{ij} \) defining \( P(S_t = j|S_{t-1} = i) \), and the output or emissions model defining \( P(y_t|S_t) \), where as mentioned before, we will use a Poisson emission distribution.

More specifically, we assume independent Poisson firing statistics for each neuron, so that the emission probability for the \( i \)th state is modeled by a spatially varying (state-dependent) multivariate Poisson process:

\[
P(y_t|S_t = i; \theta) = \prod_{n=1}^{N} P(y_{n,t}|S_t = i; \theta)
\]

\[
= \prod_{n=1}^{N} \prod_{j=1}^{Z} P(y_{n,t}|S_t = j; \theta)^{S_{t,i}}
\]

\[
= \prod_{n=1}^{N} \prod_{j=1}^{Z} \left( \frac{\exp(-\lambda_{jn})\lambda_{jn}^{y_{n,t}}}{y_{n,t}!} \right)^{S_{t,i}}
\]

where \( \theta = \{\pi, A, \Lambda\} \) are the model parameters, \( \Lambda \in \mathbb{R}^{Z \times N} \) are the tuning curve parameters (a spike firing rate \( \lambda \) for every possible state \( j \in \{1 \ldots Z\} \) for each neuron \( n \in \{1, \ldots N\} \)), and \( S_{t,i} = 1 \) iff \( S_t = i \), and 0 otherwise.

Finally, we assume that our model is time-invariant: that is, we assume that the state transition probability matrix and the state-dependent emission distributions do not change over time.

Then, given a training set \( \mathcal{D} = \{y^{(1)}_{1:T_1}, \ldots, y^{(M)}_{1:T_M}\} \), containing \( M \) sequences of observations, and since the training sequences are assumed to have been drawn
independently, the complete data likelihood takes the form

\[
P(\mathcal{D}, \mathbf{S} | \theta) = \prod_{m=1}^{M} P\left(\mathbf{y}^{(m)}_{1:T_m} | \theta, \mathbf{s}^{(m)}_{1:T_m}\right) P\left(\mathbf{s}^{(m)}_{1:T_m}\right).
\]

The model parameters \(\theta = \{\pi, A, \Lambda\}\) can then be estimated using standard methods including Expectation Maximization (EM), Variational Bayes (VB), or Monte Carlo methods. We opted for the fast and efficient EM algorithm to learn the parameters in our models (see Rabiner, 1989).

The typical data processing workflow is shown in Figure 3.1, where data are recorded using extracellular multichannel electrodes (specifically tetrodes for all of our recordings), after which spike detection and spike sorting are performed. The sorted spikes are then binned into 20 ms bins, and the vector of spike counts within such a bin then constitutes a single observation, \(\mathbf{y}_t\).

![Figure 3.1: Data preprocessing workflow for switching Poisson hidden Markov models of neural activity.](image)

Left: data preprocessing workflow for extracellular recordings of neural activity; extracellularly recorded multichannel neural activity is recorded, then filtered for spike detection; following spike detection, the spikes are sorted into units (putative neurons) to obtain a spike train for each unit. These spike trains are binned into non-overlapping time bins, and the numbers of spikes are counted within each bin to form the inputs to the HMM. Right: the probabilistic graphical model representation of our HMM, with example inputs on each of the observation nodes \((\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \ldots, \mathbf{y}_T)\).

More specifically, for our PBE models, we of course first restrict the neural activity to only those epochs during PBEs before binning them into 20 ms bins. Each PBE therefore had a small number of 20 ms bins associated with it, and together constituted one sequence from the HMM’s perspective. We chose 20 ms bins so that even a short 100 ms PBE would have at least five bins to show temporal dynamics, and also so that most bins contained at least one spike. If the bins are too short, then the likelihood of
observing no spikes within a bin becomes too great, and if the bin size is too long, then we don’t have sufficient resolution to capture fine temporal dynamics. Moreover, the bins are non-overlapping so that we do not introduce any artificial sequential structure (by smoothing, for example) into the data.

Figure 3.2: A hidden Markov model of ensemble activity during PBEs (reproduced from Maboudi et al., 2018).
A hidden Markov model of ensemble activity during Population Burst Events. a. Examples of three PBEs and a run epoch. b. Spikes during 7 example PBEs (top) and their associated (30 state HMM-decoded) latent space distributions (bottom). The place cells are ordered by their place fields on the track, whereas the non-place cells are unordered. The latent states are ordered according to the peak densities of the Latent State Place Fields (lsPFs, see section 3.1.3). c. The transition matrix models the dynamics of the unobserved internally-generated state. The sparsity and banded-diagonal shape are suggestive of sequential dynamics. d. The observation model of our HMM is a set of Poisson probability distributions (one for each neuron) for each hidden state. Looking across columns (states), the mean firing rate is typically elevated for only a few of the neurons and individual neurons have elevated firing rates for only a few states.

Figure 3.2 shows an example of an HMM that was fit on PBEs from an animal on a linear track. Note that the transition matrix appears to be largely banded diagonal, consistent with Markovian movement on a linear track. Note further that for visualization purposes, we wanted to order the states to maximize the super diagonal of
the transition matrix. We used a greedy approach which typically yields this solution. In particular, we start by assigning the first index to the state with the highest initial probability and keep adding states based on the most probable state transitions. For all of our linear track analyses, we used \( Z = 30 \) states, but the results are qualitatively similar for a wide range of number of states (see Maboudi et al., 2018, Figure 3-Figure supplement 1).

3.1.2.3 Model structure in PBE-derived HMM

Using separate training- and test-datasets (cross-validation) mitigates overfitting to training data, but it is still possible for the cross-validated goodness-of-fit to increase with training without any underlying dynamics, e.g., if groups of neurons tend to activate in a correlated fashion, so we may reasonably wonder:

Do our PBE-derived HMMs reflect underlying sequential structure of memory traces beyond pairwise co-firing?

To answer this question, we cross-validated the model against both real “test” data and against surrogate “test” data derived from shuffling each PBE in two ways: one in which the binned spiking activity was circularly permuted across time for each neuron independently of the other neurons (“temporal shuffle”, which removes co-activation), and one in which the order of the binned data was scrambled coherently across all neurons (“time-swap”, which maintains co-activation). Note that the second shuffle preserves pairwise correlations while removing the order of any sequential patterns that might be present. Using five-fold cross-validation, we compared learned models against both actual and surrogate test data and found that the model likelihood was significantly greater for real data (vs. temporal shuffle, \( p < 0.001 \), vs. time-swap, \( p < 0.001 \), \( n = 18 \) sessions, Wilcoxon signed-rank test (see Maboudi et al., 2018, Figure 1-Figure supplemental 1).

To further understand what structure we learn from PBE activity, we compared our HMMs (trained on real data) against models trained on multiple different surrogate datasets (see Figure 3.3). These surrogate datasets were obtained from actual data following: 1) temporal shuffles and 2) time-swap shuffles, as above, and 3) by producing a surrogate PBE from independent Poisson simulations according to each unit’s mean firing rate within the original PBEs. We briefly describe the different shuffling
approaches next.

**Temporal shuffle**

Within each event, the binned spiking activity is circularly permuted across time for each unit, independently of all the other units. This goal of this shuffle is to disrupt unit co-activation, while maintaining the temporal dynamics for each unit.

**Time-swap shuffle**

Within each event, the order of the binned columns of neural activity is randomly permuted across time, coherently across units. The goal of this shuffle is to change the temporal dynamics of ensemble activity, while maintaining unit co-activation.

**Pooled time-swap shuffle**

The order of the binned columns of neural activity is randomly permuted across all pooled events, coherently across units. This shuffle has been previously used in Bayesian replay detection (Davidson et al., 2009).

**Poisson surrogate dataset**

We first estimate each unit’s mean firing rate across all PBEs, and then produce surrogate PBEs from independent Poisson simulations according to each unit’s mean firing rate.

After fitting models to the shuffled and surrogate datasets, we investigated the sparsity of the transition matrices using the Gini coefficient across rows of the transition matrix (the so-called departure Gini coefficient, which is high if a state is likely to transition to only a small number of states, i.e., it is sparse, and low when a state is likely to transition to one of several other states, i.e., it is non-sparse). More specifically, sparsity of the transitions from individual states (departure sparsity) was measured by calculating the Gini coefficient of corresponding rows of the transition matrix (Hurley and Rickard, 2009). The Gini coefficient is a measure of how variable the values of this probability distribution are, with equality across states corresponding to a coefficient
Figure 3.3: Models of PBE activity are sparse (reproduced from Maboudi et al., 2018). We trained HMMs on neural activity during PBEs (in 20 ms bins), as well as on surrogate transformations of those PBEs. **a.** (top) The transition matrices for the actual and surrogate PBE models with states ordered to maximize the transition probability from state \(i\) to state \(i+1\). (bottom) Undirected connectivity graphs corresponding to the transition matrices. The nodes correspond to states (progressing clockwise, starting at the top). The weights of the edges are proportional to the transition probabilities between the nodes (states). The transition probabilities from state \(i\) to every other state except \(i+1\) are shown in the interior of the graph, whereas for clarity, transition probabilities from state \(i\) to itself, as well as to neighboring state \(i+1\) are shown between the inner and outer rings of nodes (the nodes on the inner and outer rings represent the same states). **b.** The observation matrices for actual and surrogate PBE models show the mean firing rate for neurons in each state. **c.** We quantified the sparsity of transitions from one state to all other states using the Gini coefficient of rows of the transition matrix for the example session in **a.** Actual data yielded sparser transition matrices than shuffles. **d.** The observation models—each neuron’s expected activity for each state—learned from actual data for the example session are significantly sparser than those learned after shuffling. **e.** Summary of transition matrix sparsity and **f.** Observation model sparsity with corresponding shuffle data pooled over all sessions/animals. (*****: \(p < 0.001\), **: \(p < 0.01\), +: \(p < 0.05\); single session comparisons: \(n = 250\) realizations, Welch’s t-test; aggregated comparisons - \(n = 18\) sessions, Wilcoxon signed-rank test).
by only a few other, in turn, sparse states, providing long sequential paths through state space—consistent with spatial relationships in the environment in which the animal was behaving, but generated from PBEs.

Next, we quantified the sparsity of the observation models, again by computing the Gini coefficient across each row of the observation matrix. We found that actual data yielded mean firing rates which were also highly sparse, indicating that individual neurons were likely to be active during only a small fraction of the states. The increased sparsity of the observation model and transition matrix in the example session was representative of a significant increase over all remaining sessions ($p < 0.05$, $n = 18$ sessions, Wilcoxon signed-rank tests).

The sparsity of the full transition matrix was calculated by averaging the Gini coefficient across rows. For analyses of PBE models from linear tracks, we computed the mean sparsity across states for each of the 250 surrogate datasets, and these means were used to generate the box plots of Figure 3.3c. Note that for the actual data, we generate a distribution by randomly initializing the model 250 times and calculating the mean sparsity over all initializations. For analyses of models learned from PBEs in open fields (and the linear track comparison with 50 states), we created 50 surrogates/random initializations. Finally, to compare across sessions, we calculated the mean sparsity by averaging over all 250 surrogate datasets to obtain a single mean sparsity per session, so that $n = 18$ per-session means were used to create the box-plots of Figure 3.3e. As with transitions, we calculated mean sparsity across units for each surrogate dataset, and we then averaged over all surrogate datasets to obtain a per-session average, used in Figure 3.3f.

### 3.1.3 Hidden Markov models from PBEs capture the place code

The previous sparsity observations indicate that PBEs inform an HMM about extant spatial relationships within the environment. So, naturally, we asked:

**How do the firing patterns of neurons during actual behavior project into the learned latent spaces?**

To observe the evolution of the latent states during behavior, we used our model to determine the most likely sequence of latent states corresponding to neural activity observed in 100 ms bins during epochs that displayed strong theta oscillations (exclusive
Hidden Markov models for exploratory data analysis

Chapter 3

of PBEs) when rats were running (speed > 10 cm/s). If the learned model was distinct from ensemble patterns during behavior, we might expect the resulting state space probability distributions at each point in time to be randomly spread among multiple states. Instead, we found distributions that resembled sequential trajectories through the latent space (see Figure 3.4a), closely resembling the physical trajectories made by the animal along the track, further demonstrating that the latent state dynamics learned from PBEs correspond to an internalized model of physical space.

Figure 3.4: Latent states capture positional code (reproduced from Maboudi et al., 2018).

a. Using the model parameters estimated from PBEs, we decoded latent state probabilities from neural activity during periods when the animal was running. An example shows the trajectory of the decoded latent state probabilities during six runs across the track. b. Mapping latent state probabilities to associated animal positions yields latent-state place fields (lsPFs) which describe the probability of each state for positions along the track. c. Shuffling the position associations yields uninformative state mappings. d. For an example session, position decoding during run periods through the latent space gives significantly better accuracy than decoding using the shuffled tuning curves. The dotted line shows the animal’s position during intervening non run periods. e. The distribution of position decoding accuracy over all sessions (n = 18) was significantly greater than chance. (p < 0.001).

To better understand the relationship between the latent space and physical space, we used the latent state trajectories decoded during running (like those shown in
Figure 3.4a) to form an estimate of the likelihood of each state as a function of location on the track. These “Latent State Place Fields” (lsPFs, Figure 3.4b) in many ways resemble neuronal place fields and similarly tiled the extent of the track. This spatial localization went away when we re-estimated the lsPFs with shuffled positions (Figure 3.4c). To quantify how informative the latent states were about position, we used the lsPFs to map decoded state sequences to position during running periods (Figure 3.4d). In our example session, decoding through the latent space resulted in a median decoding accuracy of 5 cm, significantly greater than the 47 cm obtained from shuffled lsPFs ($p < 0.001$, Wilcoxon signed-rank test, Figure 3.4d).

Figure 3.5: Decoding accuracy for different numbers of states.

a. We computed the median position decoding accuracy (via the latent space) for each session on the linear track (n = 18 sessions) using cross validation. In particular, we learned a PBE model for each session, and then using cross validation we learned the latent space to animal position mapping on a training set, and recorded the position decoding accuracy on the corresponding test set by first decoding to the state space using the PBE model, and then mapping the state space to the animal position using the lsPF learned on the training set. The position decoding accuracy was significantly greater than chance for each of the 18 sessions ($p < 0.001$, Wilcoxon signed-rank test). 

b. For an example session, we calculated the median decoding accuracy as we varied the number of states in our PBE model (n = 30 realizations per number of states considered). The decoding accuracy is informative over a very wide range of number of states.

c. For the same example session, we show the lsPFs for different numbers of states. The lsPFs are also informative over a wide range of number of states, suggesting that our analyses are largely insensitive to this particular parameter choice.
When we evaluated the decoding error over our full set of sessions, we observed a similar result ($p < 0.001$, Wilcoxon signed-rank test, Figure 3.4e). As our method requires discretizing the state space, a potential caveat is that the number of latent states is a relevant (and perhaps a sensitive) parameter, which we arbitrarily chose to be 30. However, latent-state place fields were informative of position over a wide range of values of this parameter (Figure 3.5). Note that decoding into the latent space and then mapping to position resulted in slightly higher error than simply performing Bayesian decoding on the neural activity during behavior. This suggests that the latent space we learn from PBEs may not capture all the information about space that is present in hippocampal activity during behavior, though this may also reflect the limited number of PBEs from which we were able learn.

\textbf{lsPFs vs place fields}

It is important to note that even though \textit{lsPFs} look qualitatively similar to place fields, there is no one-to-one mapping that would really make sense. Even if we choose the number of states to equal the number of place cells, the interpretation should be subtly different: place fields (and place cells) are used to decode position \textit{from population activity}, meaning that many place cells can be active at the same time, whereas \textit{lsPFs} show the spatial distributions over space for each state, but our system can only be in one particular state at any given time! Consequently, a slight over specification for the number of states (compared to the number of place cells) could make sense, and would improve our spatial resolution in the state space.

3.1.4 HMM-congruent PBEs capture sequence replay

It has previously been described how the pattern of place cell firing during several PBEs recapitulates the order in which they are active when animals run on the linear track (Skaggs and McNaughton, 1996; Diba and Buzsáki, 2007). Given that we use all PBEs for model learning and our models capture the structure of the environment and the patterns expressed by place cells during exploration, we were interested in understanding whether we could also use our latent-space models to find these replay events. Indeed, for many events when we decode trajectories through state space, they resemble the sequential patterns observed when we decode position using Bayesian
Chapter 3 Hidden Markov models for exploratory data analysis

techniques and the place cell map (Figure 3.6b, left). However, given previous evidence for replay of environments not recently experienced (Gupta et al., 2010; Karlsson and Frank, 2009), we hypothesized that some PBEs might contain ensemble neural activity which is unstructured and thus unrelated to the learned model, and that these would correspond to the “non-replay” events found using traditional methods.

First, we identified hippocampal replay using the traditional, versatile and widely-used Bayesian decoding method to ascribe a replay score to sequential patterns during PBEs. Briefly, for each PBE, we used place-field maps to estimate a spatial trajectory (an \textit{a posteriori} distribution of positions) in 20 ms bins. We generated surrogate data via a column-cycle shuffle (i.e., a circular shift across positions for each time bin (Davidson et al., 2009)) of the \textit{a posteriori} distributions during PBEs. The real and surrogate trajectories were scored following the approach by Davidson et al. (2009), and we defined replay events as those for which the score of the actual trajectory was larger than a threshold fraction of the null distribution generated by the surrogate scores. Using this approach, we found that 57% of PBEs (1064 of 1883) were identified as replay beyond a threshold of 99% (median across datasets 54.2%, inter quartile range = 32.8–61.0%). Thus, as has been reported many times (Davidson et al., 2009; Diba and Buzsáki, 2007; Foster and Wilson, 2006; Karlsson and Frank, 2009), only a fraction of PBEs (but many more than expected by chance) represent statistically significant replay.

\begin{itemize}
  \item \textbf{HMM congruence as a proxy for sequence replay}
  \end{itemize}

Instead of trying to find the same set of significant replay sequences described above, we want to identify (test-) sequences that are \textit{congruent} with our (trained) HMM. These sequences can be thought of as strongly expressed sequences in the model, and we hypothesized that such congruent sequences are likely to be found repeatedly in our training set (but interestingly---and in contrast to the traditional Bayesian decoding approach or template matching approaches---not necessarily during animal behavior).

To assess how well the pattern of ensemble activity during individual PBEs related to the overall state-space model learned from PBE activity (“congruence”), we developed a statistical approach for identifying the subset of strongly structured
PBEs. Specifically, rather than comparing real and surrogate PBEs, we compared the goodness-of-fit for each event to a null distribution generated via a computationally-efficient manipulation of the transition matrix of the model (see Figure 3.6b); we row-wise shuffled the non-diagonal elements of the transition matrix to assess whether an individual PBE is a more ordered sequence through state space than would be expected by chance. Maintaining the diagonal avoids identifying as different from chance sequences which consist of few repeated states, marked by transitions between state $i$ and itself.

The fraction of events identified as replay using Bayesian decoding is strongly tied to how the null-distribution is generated (i.e., what shuffle is used), some secondary criteria (e.g., number of active cells, unit cluster quality, peak firing rate, trajectory “jumps”, etc.), and the value of the significance threshold arbitrarily chosen to be 90%, 95%, or 99% of shuffles in different reports. When we combined across datasets, we found that our transition matrix shuffle yielded a null distribution for which a 99% confidence interval identified slightly fewer PBEs as significant than the column-cycle shuffle did for Bayesian decoding (Figure 3.6c).

To make a principled comparison of Bayesian- and HMM-based replay detection schemes, we fixed the Bayesian-based significance threshold at 99% but selected the significance threshold for the HMM-congruence null distribution so that the fraction of replay events detected would be the same between the two schemes. Following this approach, we found that model-congruent/incongruent PBEs largely overlapped with the replay/non-replay events detected using Bayesian decoding of the place cell map (Figure 3.6d). Thus, using only the neural activity during PBEs, without access to any place cell activity, we are remarkably able to detect the sequential patterns typically described as “replay” based only on their consistency with the structure of other PBE activity.

There were, however, also differences between the Bayesian and HMM-congruent approaches, including events that reached significance in one but not the other formalism. We wanted to understand where and why these approaches differed in identifying significant sequences. When we examined individual PBEs, we found sequences for which both Bayesian and model-congruence replay detection approaches appeared to malfunction (Figure 3.7a).
Figure 3.6: Sequence replay detection using HMMs (from Maboudi et al., 2018).

a. Example PBEs decoded to position using Bayesian decoding. b. (left) Same examples decoded to the latent space using the learned HMM. (right) Examples decoded after shuffling the transition matrix, and (middle) the sequence likelihood using actual and shuffled models. c. Effect of significance threshold on the fraction of events identified as replay using Bayesian decoding and model congruent events using the HMM approach. d. Comparing Bayesian and model-congruence approaches for all PBEs recorded, we find statistically significant agreement in event identification (60.9% agreement, n = 1883 events from 18 sessions, p < 0.001, Fisher’s exact test two sided).

These differences were not a failure of the choice of significance threshold, as for both techniques we found what appeared to be false-negatives (patterns which looked like replay but were not significant) as well as false-positives (patterns which looked noisy but were identified as significant). Thus, in order to quantitatively compare the two approaches, we asked eight humans to visually examine all the PBEs in our database. They were instructed to label as replay PBEs in which the animal’s Bayesian decoded position translated sequentially without big jumps (similar to the criteria used by Silva et al., 2015).

We marked each event as a “true” community replay if it was identified by a majority of scorers (six individuals scored n = 1883 events, two individuals scored a subset of n = 1423 events, individual scores are shown in Figure 5-Figure supplement 1 of Maboudi et al. (2018)). We calculated an ROC curve which compared the rate of true positive and false positive detections as the significance thresholds for Bayesian and model-congruence approaches were varied (Figure 3.7b). A perfect detector would have an Area under the curve (AUC) of unity. We did not find a significant difference between the AUCs of Bayesian decoding and model-congruence (p = 0.14, using
Figure 3.7: Bayesian replay detection vs HMM congruence (reproduced from Maboudi et al., 2018).

a. Sixteen examples from one session show that Bayesian decoding and HMM model-congruence can differ in labeling of significant replay events. For each event, spike rasters (ordered by the location of each neuron’s place field) and the Bayesian decoded trajectory are shown. “+” (“−”) label corresponds to significant (insignificant) events. (left) Both methods can fail to label events that appear to be sequential as replay and (right) label events replay that appear non-sequential.

b. We recruited human scorers to visually inspect Bayesian decoded spike trains and identify putative sequential replay events. Using their identifications as labels, we can define an ROC curve for both Bayesian and HMM model-congruence which shows how detection performance changes as the significance threshold is varied. (inset) Human scorers identify 24% of PBEs as replay. Setting thresholds to match this value results in agreement of 70% between Bayesian and HMM model-congruence.

c. Using the same thresholds, we find ≈ 70% agreement between algorithmic and human replay identification. (All comparison matrices, \( p < 0.001 \), Fisher’s exact test two-tailed.)

bootstrap resampling). If we select thresholds such that our algorithms yield a similar fraction of significant vs. total events as the 24% denoted by our human scorers, we find that both Bayesian and model-congruence yield an agreement of ≈ 70% labeled events with each other and with human scorers (Figure 3.7c).

Thus, congruence with an HMM trained only on PBEs appears to work as reliably
as Bayesian decoding in detecting sequential reactivation of linear track behaviors. However, when we examined individual sessions, we noticed that performance was quite variable. Given that our models are learned only from PBEs, we reasoned that the statistics or structure of the PBEs within each session might yield models which vary in quality depending on the number of recorded units, the number of PBEs detected, and their self-consistency across events. We created a model quality metric by comparing cross-validated learning statistics to models which were learned from shuffled events. We found that the performance of model-congruence detection was tied to model quality \((R^2 = 0.17, F = 2.9, n = 18 \text{ sessions})\), and that model quality, in turn, was highly correlated with the number of PBEs during the session \((R^2 = 0.96, F = 392.6, n = 18 \text{ sessions})\). Not surprisingly, the performance of Bayesian decoding relative to human scorers was independent of the quality of the HMM, or the number of PBEs, as the place field model is learned from ensemble neural activity during behavior. Thus, we find an intriguing contrast—when there is an abundance of PBEs (indicating novelty, learning, hippocampus-dependent planning, etc.(Buzsáki, 2015)), even in the absence of repeated experience, replay detection based on PBE activity is highly effective. Conversely, when there are few PBEs (i.e., scenarios in which PBEs are uncorrelated with cognitive function), but an abundance of repeated behavioral trials, Bayesian decoding of these limited events proves more effective.

It should be mentioned that our session quality metric that we presented in Maboudi et al. (2018) is not sufficient to determine if a dataset contains sequentially structured data. As an example, if we have a (rather uninteresting) state sequence of \((a, a, \ldots, a, b, b, \ldots, b)\) indicating very slow (almost no-) temporal dynamics, then comparison with a time’swap shuffle will suggest that we have completely destroyed the temporal dynamics, and hence that the original sequence was likely to have been “sequentially interesting”. A more robust session quality metric is therefore desired, but the one we have developed can nonetheless provide some insight, and can identify several classes of sequentially-random datasets for which the HMM approach would be entirely unsuitable. Our session quality metric therefore gives us a necessary, but not a sufficient condition for knowing if further analyses with our HMM approach might be warranted or justified.
3.1.5 HMMs for unconstrained environments

Most previous reports of replay have either used linear tracks (see e.g., Diba and Buzsáki, 2007) or simple mazes that can be easily be linearized (such as the w-maze used by Karlsson and Frank, 2009). The primary reason for this is that classical approaches to replay detection, including Bayesian decoding, template matching and order statistics, all require linear(ized) templates. One notable exception is the work by Pfeiffer and Foster (2013, 2015); Silva et al. (2015), where a number of ad-hoc empirical rules were specified to characterize two-dimensional Bayesian decoded paths as trajectories or not. These rules included minimum end-to-end travel distance, and maximum jump discontinuities between consecutive frames, and so on. Indeed, template matching works well in simple linear environments, but even in an open field, there are infinitely many trajectories (and hence templates) that the animal can take. Memories more generally can also not reasonably be expected to fit within the constraints of a linearized representation.

The HMM approach to sequence detection that we presented in section 3.1.4 is essentially agnostic to the underlying representational complexity. Indeed, the transition matrix captures the dynamics between all pairwise state transitions, allowing for arbitrary movement through the state space. Of course, if certain transitions are unlikely or impossible due to the topology of the environment, those same restrictions will be evident in the transition probabilities in the transition matrix. The only issue is that we may need significantly more data to learn robust dynamics in less-constrained environments, since there are many more possible paths to explore through the state space.

To demonstrate the efficacy of our HMM approach to generalize to more complex environments and behavioral tasks, we considered data from CA1 neurons in rats as they explored in a 2 m × 2 m open field arena for liquid reward (Pfeiffer and Foster, 2013, 2015). Briefly, animals were trained to discover which one of 36 liquid reward wells would be the “home” well on a given day. They were then required to alternate between searching for a randomly rewarded well and returning to the home well. Using the place cell map in this task and Bayesian decoding, many PBEs were decoded.

It is entirely agnostic to the underlying representational complexity, assuming that we have sufficiently many training examples to fit the model with.
to trajectories through two-dimensional space that were predictive of behavior and shaped by reward.

Figure 3.8: Modeling PBEs in open field (reproduced from Maboudi et al., 2018).  
**a.** The transition matrix estimated from activity detected during PBEs in an example session in the open field.  
**b.** The corresponding observation model (203 neurons) shows sparsity similar to the linear track.  
**c.** Example latent state place fields show spatially-limited elevated activity in two dimensions.  
**d.** For an example session, position decoding through the latent space gives significantly better accuracy than decoding using the shuffled latent state place fields.  
**e.** Comparing the sparsity of the transition matrices (mean Gini coefficient of the departure probabilities) between the linear track and open field reveals that, as expected, over the sessions we observed, the open field is significantly less sparse ($p < 0.001$), since the environment is less constrained.  
**f.** In contrast, there is not a significant difference between the sparsity of the observation model (mean Gini coefficient of the rows) between the linear track and the open field. Note that the linear track models are sparser than in Figure 3.3 due to using 50 states rather than 30 to match the open field.

Using this same dataset, we trained a HMM on neural activity during PBEs in the open field. Here, we used the same PBEs detected previously (Pfeiffer and Foster, 2013, 2015) which occupied an average of $2.53 \pm 0.42\%$ of the periods during which
animals were behaving ($77.91 \pm 21.16$ s out of $3064.86 \pm 540.26$ s). Given the large number of units available in this dataset and the increased behavioral variability in the open field environment compared to the linear track, we chose to estimate HMMs with 50 latent states.

The transition matrix and observation model from a sample session are shown in Figure 3.8a,b. Despite the complex and varied trajectories displayed by animals, the HMM captured sequential dynamics in PBE activity, as in the 1D case. When we compared learned models against both actual and surrogate test data, we found that the model likelihoods were significantly greater for real data ($p < 0.001$, Wilcoxon signed-rank test).

3.1.5.1 Decreased departure sparsity reflects underlying behavioral variability

In the case of the linear track, we linked sparsity of the transition matrix to the sequential nature of behaviors in that environment. An unconstrained, two-dimensional environment permits a much richer repertoire of behavioral trajectories. However, behavior is still constrained by the structure of space—arbitrary teleportation from one location to another is impossible. We found that learning from PBEs in the open field yielded transition matrices (Figure 3.8a) that were significantly sparser than models learned from shuffled data ($p < 0.05$, Wilcoxon signed-rank test, $n = 8$ sessions). However, consistent with increased freedom of potential behaviors, when we compared the sparsity of models learned from open field PBEs with 50-state models learned from PBEs in linear tracks, the open field transition matrices were significantly less sparse (Figure 3.8e; $p < 0.001$, Mann–Whitney U test comparing 8 2D and 18 1D sessions).

3.1.5.2 Observation sparsity is similar between 1D and 2D environments

Likewise, when we examined the observation model for the open field, we found that the activity across states for individual neurons was significantly more sparse than in models learned from shuffled data ($p < 0.05$, Wilcoxon signed-rank test, $n = 8$ sessions). However, the sparsity of linear track and open field observation models were not significantly different (Figure 3.8f; $p = 0.44$, Mann–Whitney U test).

3.1.5.3 States are localized in 2D space

Similar to the lSPFs that we’ve seen before, we can decode run data using our open field HMMs and learn a mapping from the state space to position to estimate...
2D lsPFs. Indeed, we found that the latent states corresponded with specific locations in the open field, as shown in Figure 3.8c. Moreover, we were able to decode animals’ movements with significantly greater than chance accuracy by converting decoded latent states to positions using the 2D lsPF ($p < 0.001$, Figure 3.8d). Finally, we examined model-congruency for PBEs detected in the open field. Previously, it was reported that 27.3% (815 of 2980, $n = 8$ sessions) were identified as “trajectory events” (Pfeiffer and Foster, 2015). We chose a significance threshold to match this fraction, and found that there was significant overlap between the events detected through Bayesian and model-congruence techniques ($p < 0.01$, Fisher’s exact test). Thus, an HMM of the activity during population bursts captures the structure of neural activity in two dimensional environments during complex tasks and can be used to decode events consistent with trajectories through that environment.

3.2 Remote replay detection using hidden Markov models

In addition to detecting replay-like sequences as discussed in section 3.1.4, we can also use our HMMs to detect other interesting sequence-like events. Indeed, we have already pointed out that some of the model-congruent events do not correspond to replay events identified using the Bayesian decoding approach, so here we will describe some alternatives for what those congruent-but-not-replay events might be.

More specifically, we will use our HMM approach to identify so-called “remote” replay events in section 3.2.2 using behavioral data from local and remote contexts (but still in an unsupervised sense, so that there is no position information available to the model). Then, in section 3.2.3, we will identify congruent events directly from the local PBEs and local Bayesian significant events as a way to identify likely remote (or non-spatial) events even in the complete absence of non-local recordings.

The remote replay identification from local and remote behavioral data has been published by Ackermann et al. (2017), and the non-spatial analysis was presented in Maboudi et al. (2018).

3.2.1 Awake replay of remote experiences

To demonstrate the use of HMMs for the identification of remote replay events, we consider the experiment presented by Karlsson and Frank (2009), where it was
shown that rats exhibit robust replay of remote experiences during awake periods\(^2\).

![Remote replay task illustration](image)

Figure 3.9: Remote replay task illustration (reproduced from Ackermann et al., 2017). Each day of the experiment, rats were exposed to a series of contexts, including two distinct w-shaped mazes, Environment 1 (E1) and Environment 2 (E2). Remote replay was defined as robust representations of E1 while the animal was running in E2.

Briefly, ensembles of pyramidal cells were recorded from hippocampal areas CA1 and CA3 while three rats were exposed to two w-shaped environments (Figure 3.9 E1 and E2), in which the rats were rewarded at the endpoint of each arm when correctly performing a continuous alternation task. Rats were exposed to E2 for several days before the first exposure to E1, so that E2 was always more familiar to the animals than E1. Remote replay (more specifically, awake remote replay) was then defined as robust representations (during SWRs, not the closely-related PBEs that we have used thus far) of E1 while the animal was in E2, or robust representations of either E1 or E2 when the animal was awake and in the rest box. We will only consider remote replay of E1 while the animal was in E2 here.

### 3.2.1.1 Regression analysis for remote replay analysis

Following Karlsson and Frank (2009), candidate remote replay events were identified as SWRs (recorded while the animal was in E2) during which at least 5 neurons that had place fields in E1 were active. Candidate events were then divided into 15 ms bins, and a Bayesian decoder with a uniform prior was used to decode the ensemble neural activity to distributions over positions in environment E1.

Critically, the position was first linearized in one of two ways: (i) for events where the trajectory went to or from the center arm, linearized position was defined as the distance from the reward well on the center arm, and (ii) for events that had trajectories that went from one outer arm to the other, linearized position was defined as the distance from the upper left reward well.

\(^2\)The data is publicly available from [crcns.org](http://crcns.org).
Figure 3.10: Examples of linearized replay detection using regression analysis (reproduced from Ackermann et al., 2017).

(a) and (c) Examples of remote replay candidate events that were correctly classified as significant remote replay by the regression-based analysis. (d) An example of a candidate remote replay event that was correctly classified as non-significant. (b) An example of a remote replay event that was misclassified as non-significant because it exhibited nonlinear trajectory behavior. In each panel, the spike raster is shown on the left, with the decoded linearized position distribution for each time bin shown on the right, along with a cartoon representation of the decoded trajectory in the w-maze.
Candidate events were then scored by determining the $R^2$ value from a regression of (decoded and linearized) position over time, compared to 10,000 regressions on surrogate events where the order of the time bins were randomly permuted. A $p$ value for each event was then calculated as the proportion of shuffled $R^2$ values greater than the actual $R^2$ value, and an event was considered significant when $p < 0.05$.

Figure 3.10 shows four example candidate remote replay events, along with the spike rasters for all the place cells in E1, and the decoded distributions over linearized position for each time bin. Time bins with no spikes cannot be meaningfully decoded to position, and were not included in the regression analysis. Fig. Figure 3.10.a and Figure 3.10.c show clear linear movement through the environment, and both were correctly identified as remote replay by the regression analysis.

### 3.2.1.2 Limitations of regression analysis

Even though this regression-based analysis is effective for finding examples of remote replay in the w-maze task, it critically relies on (i) the linearization of position, and on (ii) the position data being available. It is non-trivial, for example, to extend this analysis to less constrained or more complex environments (such as open fields), or to other behavioral correlates that are not easily observable (such as sequences of odor cues, or other sequences of episodic memory).

It is with these more complex tasks in mind that we set out to develop an approach that (i) generalizes trivially to more complex environments and tasks, and (ii) that can work in the absence of observable behavioral correlates. We will demonstrate that the HMM can be used to achieve both of these goals.

### 3.2.2 Hidden Markov models for remote replay detection

We are primarily interested in determining when short bursts of neural activity during SWRs encode remote experiences or environments. To this end, we searched for evidence of sequences corresponding to environment 1 (E1) while the animal was running in environment 2 (E2), as reported in Karlsson and Frank (2009). For each animal, and for each experiment day, we learned two HMMs (one for each environment), using bouts of run activity (animal speed > 3 cm/s), while excluding any SWRs. We subsequently scored all the candidate remote replay events recorded in E2, in the HMM corresponding to E1. Scoring SWR-associated sequences from E2 in the HMM from E1 allows us to determine if the sequences are consistent with those observed
during active behavior in E1.

More specifically, we learned the HMMs using a time bin size of $\Delta t = 125$ ms, which is (i) short enough to capture the behavioral dynamics of the animals while running, and moreover (ii) it captures a full theta cycle ($\approx 8$ Hz in rodents). We arbitrarily chose $m = 30$ states for our models, but we have previously shown that the analyses are remarkably insensitive to the actual choice of the number of states.

Similar to the regression analysis performed in Karlsson and Frank (2009), we employed a time swap shuffle to determine congruence with our HMMs. In particular, for each candidate event, we synthesized $10,000$ surrogate events by randomly permuting the observation time bins, and scoring each surrogate event in the E1-HMM to form a shuffle distribution for that event. The $p$ value for the event was then defined as the proportion of the shuffled scores that was greater than the score of the actual event. (see e.g., Figure 3.12 for some example shuffle distributions and the scores of the actual events). Significant events were defined as those that had $p < 0.05$, as in Karlsson and Frank (2009).

Note that the behavioral-timescale HMMs were learned with $\Delta t = 125$ ms, but the candidate SWR-associated events were binned into $\Delta t = 15$ ms bins, as in Karlsson and Frank (2009). For all the results presented here, we did not apply any scaling to either the events, nor the model, when scoring the candidate events. Moreover, we found that the results were largely insensitive to any scaling that we did try (results not shown).

We performed both the regression-based analysis from Karlsson and Frank (2009) as well as our HMM-based analysis for all animals, and all experiment days (see Table 3.2), but we show results for only a single representative session (rat 2, day 3) here.

### 3.2.2.1 HMMs capture the remote positional code

Since our HMMs are learned during running behavior, it is natural to expect that the latent states should somehow encode position in their abstract representations, even more so than those models from PBEs that we showed in section 3.1.3. Indeed, following the same approach as before, we computed the lsPFs of E1, as shown in Figure 3.11
We decoded neural activity during running behavior to the state space, and learned a mapping from the state space to physical position. The likelihood of the animal’s position is shown for each of the $m = 30$ states in our HMM from E1.

### 3.2.2.2 HMMs can identify examples of remote replay

Following Karlsson and Frank (2009), we identified SWR-associated candidate events for remote replay in E2, and scored all of these events in our E1-HMMs. In our example session, and using the significance threshold of $p < 0.05$, we identified 26 out of 36 candidate events as significant remote replay (72%), in comparison to 23 out of 36 (64%) using the regression-based analysis (see Table 3.1). Overall, there is a 69% agreement between the two approaches (19 + 6 out of 36).

<table>
<thead>
<tr>
<th>$N$</th>
<th>$R^2$ +</th>
<th>$R^2$ −</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM +</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>HMM −</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 3.1: Regression vs HMM-based remote replay detection.
Comparison between the number of significant remote replay events obtained using regression-based and our HMM-based approaches. Here, $R^2$ denotes the regression-based analysis, ‘+’ denotes significant events, and ‘−’ denotes non significant events.
3.2.2.3 HMMs can identify nonlinear examples of remote replay

Closer inspection of the remote replay events (see e.g., Figure 3.12) reveals some interesting differences between the two approaches. In particular, the regression-based analysis requires the position on the w-maze to be linearized (which introduces a position ambiguity in one of the arms), and then requires remote replay events to be linear traversals in this linearized space. However, using our HMM approach,
which is learned completely independent of the position data, we do not need to concern ourselves with the linearization, and moreover, we can identify nonlinear remote replay events where the trajectory might back-track or exhibit some other interesting-but-consistent behavior.

One example of such a nonlinear remote replay event that was correctly classified by the HMM-based approach (and misclassified by the regression analysis) is shown in Figure 3.12.b. Figure 3.12 also shows three other examples of remote replay candidates, two of which are correctly classified as significant remote replay (Figure 3.12.a and c) and one that was correctly classified as non significant (Figure 3.12.d). Notice that in all the examples shown, the candidate events are nonsensical in the current (local) environment, E2, where data is recorded from.

The HMM also seemed to have slightly fewer false positives (than the regression-based analysis), or at least to be more selective in the session considered here. For example, Figure 3.13 shows an example event that was classified as significant by the regression analysis, while being rejected by the HMM. However, a comprehensive analysis of classification accuracy (which would necessarily again have to rely on subjective determinations by human scorers) has not been carried out yet.

Results pooled across all animals, and all experiment days, are summarized in Table 3.2. In particular, we found 188 out of 623 (30%) significant remote replay events using our HMM approach, and 270 out of 623 (43%) using the regression analysis. The overall agreement between the two approaches is therefore 69% (131 + 296 out of 623), similar to the agreement between Bayesian significant replay events.
Table 3.2: Pooled comparison of regression vs HMM detection of remote replay. Comparison between the number of significant remote replay events obtained using regression-based and our HMM-based approaches for all sessions and animals combined. Overall agreement is 69% ($p < 0.001$, Fisher’s exact test, two-tailed).

<table>
<thead>
<tr>
<th></th>
<th>$R^2 +$</th>
<th>$R^2 -$</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM +</td>
<td>131</td>
<td>57</td>
</tr>
<tr>
<td>HMM -</td>
<td>139</td>
<td>296</td>
</tr>
<tr>
<td></td>
<td>270</td>
<td>353</td>
</tr>
</tbody>
</table>

3.2.3 Non-spatial and unobserved remote contexts

So far, we’ve shown how we could use the HMM framework to identify remote replay events, but we used the neural data (not the behavior!) from the remote context to identify those sequences. More generally, we may be interested in identifying remote events even without access to the remote neural activity. Here we develop such an approach. Such an approach can allow us to identify events from completely unobserved remote contexts, or perhaps even non-spatial memories, by excluding the local spatially-consistent events.

As described earlier, while we observed a similar fraction of events to be similar by HMM-congruence and Bayesian decoding, there was not an exact event-to-event correspondence. An intriguing potential explanation is that the latent space represented in PBE sequential firing and captured by the HMM is richer than simply the spatial structure of the present environment. In most hippocampal ensemble recording experiments, maze or open field tasks are structured to intentionally map memory elements to spatial behavior, and thus this potential richness is difficult to test.

We considered the possibility that in the awake behaving animal, PBE activity might be sequential reactivation of environments other than the one being explored (as shown in section 3.2.1). We reasoned that we could enhance the model’s representation of remote environments by filtering out local replay from the training data. We evaluated how the model-quality of our HMM changed as progressively more sequences labeled as replay by Bayesian decoding were removed from the training data.

In the linear track sessions we considered, we found that refining the training data resulted in models that lowered in quality at different rates as the threshold
for Bayesian replay was decreased (Figure 3.14). Most, but not all, models dropped precipitously in quality: > 50% when we removed events detected as Bayesian replay at a 95% threshold, as would be expected if the HMM represented only the local environment. In many outlier sessions in which model quality decreased more slowly, the initial (baseline) model quality was low. Intriguingly, however, in at least one outlier session where model quality decreased slowly with refinement (blue line, Figure 3.14a), the initial model quality was still high, and we further noted that position decoding using lSPFs yielded relatively high error (blue dot, Figure 3.14b). Thus, we wondered whether this and similar linear track sessions might have contained non-local or extra-spatial PBEs that were captured by the HMM.

Figure 3.14: Stable model quality after removing local spatially-consistent events suggests remote or non-spatial replay (adapted from Maboudi et al., 2018).

We trained and evaluated HMMs on the events that were not Bayesian significant (residual events) to identify potential extra-spatial structure. a. The normalized session quality drops as local-replay events above the Bayesian significance threshold are removed from the data. Each trace corresponds to one of the 18 linear track sessions, with the stroke width and the stroke intensity proportional to the baseline (all-events) session quality. The blue line identifies a session in which model quality drops more slowly, indicating the potential presence of extra-spatial information. The reduction in session quality for a W maze experiment with known extra-spatial information is even slower (green). When, instead, Bayesian-significant remote events are removed, rapid reduction in session quality is again revealed (red). b. The lSPF-based median decoding errors are shown as a function of baseline session quality for all 18 linear track sessions. The blue dot indicates the outlier session from panel a with potential extra-spatial information: this session shows high decoding error combined with high session quality. Session quality of the W maze session is also indicated on the x-axis (decoding error is not directly comparable).

In order to validate the concept of model-training refinement, we again considered the dataset from section 3.2.1 in which multiple environments were explored on the same day and remote replay was previously observed (Karlsson and Frank, 2009).
Figure 3.15: Locally-congruent but Bayesian non-significant PBEs corresponding to remote replay events (adapted from Maboudi et al., 2018).
Two example HMM-congruent but not Bayesian-significant events from the W maze session are depicted to demonstrate the fact that PBE model congruence can correspond to remote replay. a, e. Decoded trajectories in remote (top) and local (bottom) environments. b, f. Spike rasters, with place cells highlighted (blue cells are remote cells, red cells are local). c. Example state sequence decoded with locally-trained HMM. g. Bayesian decoded trajectory using local place cells. d. example event score from HMM indicating that it is a congruent event, even though h. local representation makes no sense. i–p. Same as above, but for another example event.
When we refined this model by removing Bayesian-significant local replay events from the training data, we found that the model quality decreased comparatively slowly (Figure 3.14a, green line), indicating that the HMM was capturing more than the local spatial structure. In contrast, when we used place fields from E1 to identify Bayesian-significant remote replay events and removed these from the training data, we found that the model quality decreased rapidly as with the general linear track cases (Figure 3.14a, red line). This is not altogether surprising, since we already knew that this particular dataset had many remote replay events. However, it is encouraging that we can gain some insight into the composition of events, even without access to the remote experiences.

Indeed, if we imagine that in this experiment data were only recorded during exploration of the familiar environment, classical Bayesian decoding would treat these events as noise, as shown in the bottom half of the two examples of Figure 3.15. In contrast, our HMM-based analysis finds these events to be significant, as shown in the top half of the two examples. Thus, by combining classical Bayesian decoding and HMM-congruence, we are able to identify a signature of when a HMM trained on PBEs captures sequential structure distinct from that dictated by the local environment. Additionally, in these cases, we show that specific non-local reactivation events can be identified.

3.3 Discussion

Increasing lines of evidence point to the importance of hippocampal ensemble activity during PBEs in guiding on-going behavior and active learning. Despite being the strongest output patterns of the hippocampus, however, this activity has been assumed to be interpretable only in the context of other theta-associated place cell activity expressed during behavior. Our findings demonstrate that over the course of a behavioral session, ensemble activity during PBEs alone is sufficient to form a model which captures the spatial relationships within an environment. This suggests that areas downstream of the hippocampus might be able to make use solely of PBE activity to form models of external space. In an extreme view, place cell activity might merely subserve the internal mechanisms in the hippocampus which generate PBE sequences. To the extent that animals might wish to use the spatial code obtained from PBEs to identify their current location, we show that this can be done after
translating ensemble activity into the latent states of the model.

When we examined the transition matrices we learned from PBEs, we found that they were marked by significant sparsity. This sparsity results from the sequential patterns generated during PBEs. Latent variable models have previously been used to analyze the structure of hippocampal place cell activity (Chen et al., 2012, 2014b; Dabaghian et al., 2014). In these studies, the learned transition matrices were mapped to undirected graphs which could be analyzed using topological measures. It is intriguing that similar structure is apparent in PBE activity. For example, we observed that transition matrices learned from PBEs associated with linear track behavior were significantly sparser than those learned from the open field, which we hypothesize is a consequence of the greater freedom of behavior in the latter (a topological difference). Whether hippocampal PBE activity must always be sequential, i.e., evolve through a sparsely-connected latent space, is an open and interesting question, as are differences between the latent state space dynamics learned during PBEs and those learned from place cell activity.

3.3.1 Graded, non-binary replay detection

Remarkably, evaluating the congruence or likelihood of test data against our HMM provided a highly novel method to detect events that are consistent with replay, without a need to access the “play” itself. In the process of evaluating the potential of HMMs for detecting replay, we developed an approach to compare different replay-detection strategies. Our results highlight how the data does not readily admit to a strict separation between “replay” and “non-replay” events. While it is possible that with additional shuffles or other restrictions (Silva et al., 2015), automated performance might be rendered closer to human-labeling, even human scorers had variation in their opinions. This calls into doubt judgments of memory-related functions which build on a binary distinction between replay and non-replay sequences. Model congruence, either as a raw statistical likelihood or weighted against a shuffle distribution, seems to be a very reasonable metric to associate with individual PBEs. Moreover, evaluating congruence with an HMM does not require access to repeated behavioral sequences, which may be infeasible under widely-used single- or few-trial learning paradigms or when the events involve replay of a remote environment. Given these benefits, along with computational efficiency, we would suggest that future analyses of the
downstream impact of hippocampal reactivation regress effects against this measure rather than assuming a binary distinction.

3.3.2 Learning, model congruence and replay quality

Not surprisingly, the rate of PBEs had a large effect on our ability to measure model congruence. Interestingly, it has been noted that the density of PBEs is higher during early exposure to a novel environment (Cheng and Frank, 2011; Frank et al., 2004; Kemere et al., 2013; Kudrimoti et al., 1999). This might suggest that for the animal, PBE activity could be an important source for generating models of the world when the animal is actively learning about the environment. If as hypothesized, replay is a form of rehearsal signal generated by the hippocampus to train neocortical modules (McClelland et al., 1995; Buzsáki, 1989), then indeed the brain’s internal machinery may also be evaluating whether a given sequential PBE pattern is congruent and consistent with previously observed PBEs. In later sessions, as animals have been repeatedly exposed to the same environments, downstream regions will have already witnessed many PBEs from which to estimate the structure of the world. Overall, our approach provides a novel viewpoint from the perspective of hippocampal PBEs. An interesting future line of inquiry would be to assess the extent to which a model built on PBEs during first experience of a novel environment is slower or faster to converge to the final spatial map than models built on theta-associated place activity.

3.3.3 Application to Extra-spatial Behaviors

We have analyzed data gathered in experiments in which rats carried out simple spatial navigation tasks. Thus, to some extent it is not surprising that when we decoded ensemble activity during behavior we found that spatial positions the animal is exploring are strongly associated with the latent states.

We anticipate that our approach for calculating IsPFs would be equally useful in tasks in which the hippocampal map is organized around time (Eichenbaum, 2014; Rodriguez and Levy, 2001) or other continuous variables (e.g. sound frequency (Aronov et al., 2017)). Conjunctive, non-spatial information might be one source of the apparent variability that results in many PBEs not being detected as replay using traditional Bayesian decoding. Another proposed source of this variability is reactivation of other environments. Our proof-of-concept analysis of remote replay identification using only local neural activity, however, suggests that HMMs learned
from PBEs can, in fact, capture the spatial structure of environments beyond the one the animal is currently exploring, and that it should be possible to use HMMs to infer the presence of extra-spatial sequential reactivation in PBEs as well.

3.3.4 Conclusions

We have demonstrated a new analytical framework for studying hippocampal ensemble activity which enables primacy of PBEs in model formation. We use an unsupervised learning technique commonly used in the machine learning field to study sequential patterns, the hidden Markov model. This contrasts with existing approaches in which the model—estimated place fields for the ensemble—is formed using the theta-associated place cell activity. We find that our PBE-first approach results in a model which still captures the spatial structure of the behavioral tasks we studied. Additionally, we demonstrate that we can use model-congruence as a tool for assessing whether or not individual PBEs contain hippocampal replay.

In addition, we have demonstrated how the HMMs can be used to detect remote replay without appealing to any behavioral correlates, which in contrast to existing approaches, means that our approach is largely independent of task complexity. Indeed, this flexibility makes it possible to analyze a whole new class of behaviors beyond spatial memory. Indeed, the HMM framework presents a powerful and attractive approach to analyze a variety of sequential tasks and phenomena in the hippocampus and beyond.
Chapter 4

Clusterless hidden Markov models of neural activity

“You cannot eat a cluster of grapes at once, but it is very easy if you eat them one by one.”

Jacques Roumain

We present a new model that is a variation on / extension of the standard switching Poisson hidden Markov model (where observations are spike counts from each of \(N\) neurons), to a clusterless approximation in which we observe only a \(d\)-dimensional mark for each spike. The model is semi-clusterless, in the sense that we still need to fix/infer the number of neurons, \(N\), in contrast to existing clusterless (supervised) decoding approaches for which the number of neurons do not always enter into the model.

We describe our new model in section 4.2, and present results on both simulated data (section 4.3.1) and real data (section 4.3.2). We briefly discuss related work in section 4.4, and we describe future work and remaining challenges in section 4.5.

4.1 Motivation

Spike sorting has been a standard preprocessing step to obtain ensembles of single unit data from multiunit, multichannel recording since at least the late 1970s (see e.g., Abeles and Goldstein, 1977). However, more recently, some researchers have started doing analyses directly on the unsorted data (Ventura, 2008; Kloosterman
et al., 2013). There are several reasons why spike-sorting-free (so-called “clusterless”) approaches are particularly attractive:

1. Spike sorting is tedious, challenging, and often subjective (depending on the algorithm, but people also don’t fully trust or agree on which automated approach is the best),

2. Spike sorting throws away a lot of potentially useful information; it throws away many (most) of the recorded spikes, since only the large amplitude spikes are typically sortable, and it also throws away probabilistic information when we make hard cluster / unit assignments.

3. Indeed, it has been shown that decoding performance can be improved by including those unsorted spikes directly into our analyses (Kloosterman, 2012), and this is even more true when we want to decode or analyze neural data at fine temporal time scales such as during theta sequence, or replay detection.

However, all of the clusterless approaches that have been developed so far, operate in a supervised mode, where a mapping is learned directly from the spike waveforms to the desired behavioral correlate, typically position. As we have already seen, we do not have access to behavioral correlates during replay events (or more generally, during PBEs), and our goal in this chapter is therefore to develop a model that is both unsupervised and clusterless. Such a model will then hopefully enable us to obtain the same benefits of existing clusterless approaches (including higher fidelity from the increase in data), as well as allow the types of unsupervised analyses that we have described earlier in this thesis.

Although not addressed directly here (or in other clusterless works for that matter), another internal motivation for the development of this model is the shift away from extracellular multichannel electrode recordings, to imaging modalities to record neural activity. The traces we obtain from Calcium imaging, for example, are somewhat similar to multiunit activity in that we may record flashes of activity from multiple layers of cells, and that we do not have nice, agreed-upon methods to extract spike trains from those traces—operating directly on the calcium traces would therefore be analogous to clusterless approaches for extracellular recordings.
4.2 Model specification

We present a novel clusterless HMM for analyzing and decoding multiunit sequential neural activity in an unsupervised manner. The Baum-Welch algorithm is an EM-based algorithm for estimating the maximum-likelihood parameters of an HMM (Bilmes et al., 1998), and we will use it in this chapter to do inference on our model. The Baum-Welch algorithm critically relies on our ability to evaluate $P(y_t \mid S_t)$, from which we can efficiently compute several other quantities of interest to enable inference with our models (see e.g., Rabiner, 1989). That is, we need to be able to evaluate the probability of observing a particular outcome $y_t$ for each (hidden) state $S_t$. In the following sections, we will present our full clusterless HMM, and develop a sampling-based approach to approximate $P(y_t \mid S_t)$.

The idea here is very simple. In particular, if we did know the neuron identities for each spike, then we would not have to use a “clusterless” approach, and we could instead fit a standard switching Poisson HMM. So in our clusterless model, we can similarly compute probabilities if we assume that we know the neuron identities, and we simply treat these identities as hidden or latent variables in our model. Moreover, we do not need to know exactly how many neurons there are, since a specification of the number of neurons simply determines the partitioning of our waveform feature space, so that an over-specification (or perhaps even a small under-specification) of the number of neurons should not have a significant effect on our model’s behavior. These ideas will be made precise below.

4.2.1 Notation and preliminaries

Suppose that we record from a population of $N$ neurons, with each neuron identified by $n \in \{1, 2, \ldots, N\}$. Further, suppose that we observe a $d$-dimensional mark ($m \in \mathbb{R}^d$) for each spike from a neuron. The marks could be, for example, the peak amplitudes from the four channels of a tetrode, or principal components of the observed waveform on a collection of electrodes, and so on.

If we assume that the neurons have state-dependent firing rates $r : \mathbb{Z} \rightarrow \mathbb{R}^N$ with $S_t \mapsto r(S_t) \equiv r_t$, then we consider each neuron as generating a train of events from an inhomogeneous (state-dependent) Poisson process, with an associated (state-dependent) rate, $r^n(S_t) \in \mathbb{R}^+$. To be more precise, we assume that the neurons are independent, and that they have $N$ state-dependent rates, $r_t = (r^1(S_t), r^2(S_t), \ldots, r^N(S_t))$ at time...
$t \in \mathbb{R}$, and for some state $S_t \in \{1, 2, \ldots, Z\}$. We may then collect all of these rates as a matrix $R \in \mathbb{R}_{\geq 0}^{Z \times N}$, and we will estimate these rates as part of the HMM parameter estimation process.

For simplicity, we will assume that we have only one probe (or tetrode). In general, we assume that the tetrodes are independent, so that tetrodes record from disjoint subsets of neurons, and therefore generalization to the multiprobe case is straightforward.

Now, consider a single observation window $(t - 1, t]$ simply identified with $t$, during which we observe $K(t) = K$ spike events (equivalently, we observe $K$ marks: $y_t = \{m_1, \ldots, m_K\}, m_i \in \mathbb{R}^d, i = 1, \ldots, K$), and for which we assume that each neuron has a constant firing rate for the duration of the time window. Since observation windows are conditionally independent (given the underlying states), we only need to concern ourselves with evaluating $P(y_t | S_t)$ for a single observation window (the process remains unchanged for each observation window).

Finally, let us assume that the marks (the spike waveform features) can be modeled as coming from unit-specific multivariate normal distributions, so that if some mark $m$ is generated by neuron $n$, then $m \sim \mathcal{N}(\mu_n, \Sigma_n)$. It is possible to use some other distributions here, and the choice of distribution will depend on which features are ultimately chosen to represent each mark. In this paper, we simply use the peak amplitudes on each of the tetrode channels, which can be adequately modeled as coming from unit-specific multivariate normal distributions in practice.

### 4.2.2 Sampling approach to evaluate $P(y_t | S_t)$

Recall that (to be able to use the Baum-Welch algorithm) our primary goal in this section is to find a way to evaluate

$$P(y_t | S_t = j) = P(y_t | r(S_t = j))$$

$$= P\left( (m_k)_{k=1}^{K(t)} | r_t^{(j)}, \{\mu_n, \Sigma_n\}_{n=1}^{N} \right)$$

$$= \mathcal{L}\left( r_t^{(j)} \mid (m_k)_{k=1}^{K(t)}; \{\mu_n, \Sigma_n\}_{n=1}^{N} \right)$$

(4.1)

where the dependence of the sequence of observed marks $y_t \equiv (m_k)_{k=1}^{K(t)}$ on the hidden state $S_t$ is realized through the state-dependent firing rates $r(S_t = j) \equiv r_t^{(j)}$, as well as the unit-specific cluster parameters $\mu$ and $\Sigma$. That is, conditioning on $S_t$ is equivalent
to conditioning on $r_t^{(j)}$.

Dropping the dependence of $K$ on $t$ (purely for notational simplicity), and dropping the explicit parameterization of $P(\cdot)$ by $\mu$ and $\Sigma$, we note that we desire to evaluate

$$P_{y_t, r_t^{(j)}}((m_k)_{k=1}^K | r_t^{(j)}),$$

which is hard to evaluate explicitly for two key reasons, namely (i) we do not know which neurons / units gave rise to each of the marks, and (ii) we do not know a priori how many marks we will observe in a particular observation window, so that we cannot easily specify a simple (possibly multivariate) probability distribution over the sequence of observed marks. Indeed, the dimensionality of this distribution will have to depend on how many marks we observe in each window.

To address challenge (i) above, we introduce the auxiliary hidden random variable $I^K \in \mathbb{Z}^{N \times K}$, that encodes which of the latent units gave rise to each of marks. Each column of $I^K$ is a standard unit vector $e_n$ whose elements are all zeros, except for the $n$th element, which is equal to one. That is, the $k$th column encodes the neuron identity $n$ that generated the $k$th mark. We denote this as $I^K \equiv \left( e_{u(k)} \right)_{k=1}^K$, such that $u(k) \in \mathbb{Z}$ is the neuron identity of the $k$th mark.

In particular, we note that

$$P_{y_t, I^K | r_t^{(j)}}((m_k)_{k=1}^K | r_t^{(j)}) = \int_{I^K} P_{y_t, I^K | r_t^{(j)}}((m_k)_{k=1}^K, I^K | r_t^{(j)}) \, dI^K. \quad (4.3)$$

Making the dependence on time $t$ and state $j$ implicit (i.e., letting $r \equiv r(S_t = j)$ as before), we note that

$$P_{y, I^K | r}((m_k)_{k=1}^K, I^K | r) = P_{y, I^K, r}((m_k)_{k=1}^K, I^K | r) \cdot P_{I^K | r}(I^K | r) \quad (4.4)$$

so that

$$\int_{I^K} P_{y, I^K | r}((m_k)_{k=1}^K, I^K | r) \, dI^K = \int_{I^K} P_{y, I^K, r}((m_k)_{k=1}^K, I^K, r) \cdot P_{I^K | r}(I^K | r) \, dI^K \quad (4.5)$$
and

\[
\text{LHS} = \int_{\mathcal{I}^K} P_{y|\mathcal{I}^K, r}( \{ m_k \}_{k=1}^K \mid \mathcal{I}^K, r ) \cdot P_{\mathcal{I}^K | r}( \mathcal{I}^K \mid r ) \, d\mathcal{I}^K \\
= E_{\mathcal{I}^K | r} [ P_{y|\mathcal{I}^K, r}( \{ m_k \}_{k=1}^K \mid \mathcal{I}^K, r ) ] = \text{RHS} \quad (4.6)
\]

Note that it is generally difficult to compute the integral (LHS) directly, whereas it is somewhat simpler to estimate the expected value (RHS), assuming of course, that we can sample from $\mathcal{I}^K \mid r$ according to it’s distribution.

Indeed, if we are able to sample $Q^{(i)} \sim P_{\mathcal{I}^K | r}( \mathcal{I}^K \mid r )$, where $Q^{(i)} \in \mathbb{Z}^{N \times K}$ for $i = 1, \ldots, M$, then

\[
P_{y| r}( \{ m_k \}_{k=1}^K \mid r ) = E_{\mathcal{I}^K | r} [ P_{y|\mathcal{I}^K, r}( \{ m_k \}_{k=1}^K \mid \mathcal{I}^K, r ) ] \\
\approx \frac{1}{M} \sum_{i=1}^{M} P_{y|\mathcal{I}^K, r}( \{ m_k \}_{k=1}^K \mid \mathcal{I}^K = Q^{(i)}, r ) \\
= \frac{1}{M} \sum_{i=1}^{M} P_{y|\mathcal{I}^K, r}( \{ m_k \}_{k=1}^K, K \mid \mathcal{I}^K = Q^{(i)}, r ) \\
= \frac{1}{M} \sum_{i=1}^{M} \prod_{k=1}^{K} \mathcal{N}( m_k; \mu(q^{k(i)}), \Sigma(q^{k(i)}) ) \cdot \prod_{n=1}^{N} \text{Pois}( V^{(i)}_n; r_n ) \quad (4.7)
\]

where we have assumed a multivariate normal distribution of the marks in the feature space (we could easily use a different distribution here), and where we have used the conditional independence (see Figure Figure 4.1) of the marks on the neuron identities ($Q$), and the number of observed marks ($K$) to factorize $P( \{ m_k \}, K \mid \mathcal{I}^K, r )$. Then we can finally approximate the data log likelihood simply as:

\[
\log[P_{y| r}( \{ m_k \}_{k=1}^K \mid r )] \approx \\
\approx \log \left[ \sum_{i=1}^{M} \exp \left( \sum_{k=1}^{K} \log \mathcal{N}( m_k; \mu(q^{k(i)}), \Sigma(q^{k(i)}) ) \right) \right] + \sum_{n=1}^{N} \log \left[ \text{Pois}( V^{(i)}_n; r_n ) \right] - \log(M) \quad (4.8)
\]
where, for each mark, $\mu$ and $\Sigma$ depend on the neuron identity encoded by $q^{k(i)}$, the $k$th column of the $i$th sample $Q^{(i)}$, and where $V_n^{(i)}$ is the total number of spikes from neuron $n$, for the $i$th sample. That is,

$$V_n = \sum_{k=1}^{K} u(k) = n \equiv \sum_{k=1}^{K} \left[ I^K \right]_{nk}, \quad n = 1, 2, \ldots, N,$$

so that $V_n^{(i)} = \sum_{k=1}^{K} q_n^{k(i)}$ (4.9)

and where $q_n^k$ is the $n$th element of the $k$th column of $Q$, which equals one if we assume that the $k$th mark was generated by the $n$th neuron, and zero otherwise.

We can summarize what we have so far with reference to the probabilistic graphical model shown in Figure 4.1.

**Figure 4.1: Probabilistic graphical model for the clusterless HMM.**

- $S$: discrete state, $|S| = Z$; $r$: aggregate firing rate, $\rho_n$: relative firing rate of neuron $n$; $K \sim \text{Pois}(r)$: number of marks observed; $q_k \sim \text{Multinom}(K, \rho)$: neuron identity indicator; $\mu_n$: neuron centroid; $\Sigma_n$: neuron covariance; $m_k \sim N\left(\sum_n q_n^k \mu_n, \sum_n q_n^k \Sigma_n\right)$.

First, we note that in order to sample $Q \sim P_{IK|r}(I^K | r)$, it is convenient to factorize the rate ($r$) into the aggregate rate $\sum_n r_n = \rho \parallel r \parallel_1$, and the relative rates $(\rho_1, \rho_2, \ldots, \rho_N) \equiv \rho \in \mathbb{R}^N$ s.t. $\sum_n \rho_n = 1$. In this way, the (true) rate associated with neuron $n$ is simply $r_n = r \rho_n$, and we can simply sample from the multinomial distribution of $V$, one trial at a time, for each mark identity in $Q^{(i)}$. That is, for $k = 1, \ldots, K$, sample $[Q^{(i)}]_k \sim \text{Multinomial}\left(1, \frac{r}{\parallel r \parallel_1}\right)$, where $[,]_k$ is used to denote the $k$th column. That is,

$$P_{IK|r}(I^K | r_t) \propto P_{V^K|r}(V^K | r) \sim \text{Multinom}\left(K, \frac{r}{\parallel r \parallel_1}\right) = \text{Multinom}(K, \rho). \quad (4.10)$$

Note that this sampling strategy yields the correct number of neuron identities (proportional to their rates), but it does not affect the order of the columns of $Q$.  

Department of Electrical and Computer Engineering  

57
4.2.3 Updating the state-dependent firing rates, \( r_t^{(j)} \)

Ordinarily, we may consider updating the rates according to

\[
\hat{r}_t^n \mid S_t = j = \frac{\sum_{t=1}^T \gamma_j(t)V_n(t)}{\sum_{t=1}^T \gamma_j(t)}
\]  

(4.11)

where \( \sum_{t=1}^T \gamma_j(t) \) is the expected number of times that we are in state \( j \) (or equivalently, the expected number of transitions away from state \( j \); even more explicitly, \( \gamma_j(t) = P(S_t = j \mid Y) \)), and \( V_n(t) \) is the number of marks in time window \( t \) that were generated / emitted by neuron \( n \). However, we do not know which marks were generated by which neurons, so we have to consider

\[
\hat{r}_t^n \mid S_t = j = \frac{\sum_{t=1}^T \gamma_j(t)E[V_n(t)]}{\sum_{t=1}^T \gamma_j(t)}
\]  

(4.12)

instead.

It turns out that we can efficiently compute this expectation, for time window \((t - 1, t)\), as follows:

\[
E_{T \mid r, (m_k)^{k=1}} [V_n(t)] = \sum_{k=1}^{K(t)} q_k^n(t)
\]  

(4.13)

where

\[
q_k^n(t) = \frac{\mathcal{N}(m_k^{(t)}; \mu_i, \Sigma_i)}{\sum_{n=1}^N \mathcal{N}(m_k^{(t)}; \mu_n, \Sigma_n)} \cdot \rho_k, \quad i = 1, 2, \ldots, N,
\]  

(4.14)

and where the dependence on \( t \) should be clear, namely that the sequence of marks \( (m_k) \) are those observed during time window \( t \). Note that (4.14) is computed independent of sampling\(^1\).

We can similarly derive expressions to update both \( \mu \) and \( \Sigma \), but if we assume that the features for each neuron follow a multivariate normal distribution, then we may as well use a preprocessing step such as a Gaussian mixture model (with a fixed number of neurons, or a Dirichlet process Gaussian mixture model so that we may jointly estimate the cluster parameters as well as the number of clusters) to determine the cluster parameters before fitting our clusterless HMM. In fact, in the rest of this

\(^1\)Can we do something similar where we work with the expected neuron identities instead of samples to compute \( P(m \mid r) \)? I’m not sure yet, but it does not seem trivial, although I will spend some more time looking into it...
chapter we will adopt this strategy, and we will use a simple Gaussian mixture model with a pre-specified number of hidden neurons.

4.3 Clusterless HMM Results

In this section we will fit several clusterless HMMs, to a variety of data. First, in section 4.3.1, we will consider synthetic or simulated data, which will allow us to accurately and directly evaluate the parameter estimation accuracy of our model. Next, in section 4.3.2, we will fit our model to real, multiprobe, multichannel data that were recorded from area CA1 of a rat as it traversed a linear track for liquid reward.

There are quite a few parameters that we can tweak when fitting our models, including the number of samples, \( M \), with which to approximate (4.7), the number of hidden states, the number of hidden units, the bin size, and so on. In general, the more samples we take, the more accurate our estimates will be, and we will see that over specifying the number of states and / or the number of hidden neurons typically has little to no effect on the model’s decoding ability. Nevertheless, these parameters choices can all be informed by a quick inspection of the data, as well as a careful consideration of the context within which the data were acquired.

4.3.1 Simulated data

To test our model’s ability to estimate its parameters from the data, we simulate data with known parameters. In particular, we seek to estimate (i) the transition matrix, (ii) the rate matrix, and (iii) the sequence of decoded states corresponding to some segment of the data. The data generation workflow is shown in Figure 4.2; once we have a transition matrix, we can use it to generate a state sequence of arbitrary length. We then feed this “true” state sequence into a multivariate normal mark generator, where we also provide it with the rate matrix, the cluster separation (or concentration) parameter, and the mark dimensions.

4.3.1.1 Generating synthetic data

Cluster separation It is reasonable to expect that if (neuron) cluster centroids (and the clusters themselves) are well separated, that even a naive clusterless model should be able to figure out which neuron generated which spike. The cluster separation is therefore a parameter that we want to be able to tweak and set to explore how it
After specifying the number of neurons, $N$, the number of states, $Z$, the sequence data length, $T$, the rate sparsity parameter, and the cluster separation parameter, we generate a transition matrix and a rate matrix. From the transition matrix we obtain a sequence through state space, which we then use together with the rate matrix to generate a sequence of multivariate normal observations (the mark sequence) to feed into our model.

Rate sparsity

We also used a Dirichlet distribution to specify the concentration and/or sparsity for each neuron, across states. In particular, we know from real data that neurons tend to fire in only a few locations (if they’re place cells), and analogously,
only in a few states, as we saw in all the previously shown observation matrices thus far. More specifically, for each neuron independently, we sample a distribution over states from a Dirichlet distribution with a particular concentration parameter that is shared across all neurons (the fact that it is shared is not important, but we simply chose that for simplicity). Then, we sample each neuron’s peak firing rate from a Gamma distribution, so that the firing rates for a particular neuron is the product of the normalized rates across states with the peak firing rate from the Gamma distribution.

4.3.1.2 Toy example I: small two-state system with 10 free parameters

Figure 4.3: Clusterless HMM parameter estimation and decoding on a small simulated dataset.
Top: rate maps (left: true, right: inferred). Middle: transition matrices (left: true, middle: direct estimation from true state sequence, right: inferred). Bottom: true state sequence (top), and decoded state sequence (bottom). Decoding accuracy is 97.5% on \( T = 200 \) observations (only \( T = 100 \) shown).
For the first example shown in Figure 4.3, we used $T = 200$ observations, a $d = 2$ dimensional mark space, and $M = 5000$ samples in our model. Note that the transition matrix is estimated more accurately than directly estimating it from the true state sequence, which is quite impressive, even for such a small example. Similarly, the rate map closely resembles that of the true underlying rate map, and the decoded state sequence is 97.5% accurate.

4.3.1.3 Toy example II: medium-sized system with 200 free parameters

![Clusterless HMM parameter estimation and decoding on a medium-size simulated dataset.](image)

Figure 4.4: Clusterless HMM parameter estimation and decoding on a medium-size simulated dataset.
Top: rate maps (left: true, right: inferred). Middle: transition matrices (left: true, middle: direct estimation from true state sequence, right: inferred). Bottom: true state sequence (top), and decoded state sequence (bottom). Decoding accuracy is 87.5% on $T = 2000$.

The previous example from Figure 4.3 only had $2 \times 3 = 6$ rate parameters, and
2 \times 2 = 4 \text{ transition parameters to estimate (for a total of 10 parameters). Here, as shown in Figure 4.4, we have } N = 10 \text{ neurons and } Z = 10 \text{ states, leading to } 100 \text{ rate, and } 100 \text{ transition parameters to infer (for a total of 200 parameters). Consequently we used more observations to fit the model, but the final parameter estimates (as well as the decoding accuracy) all look qualitatively good. The decoding accuracy in this particular example was however only 87.5\%, but considering that the chance level is around 10\%, it is clear that the model has learned a significant amount of information from the data. We could potentially improve the performance further by increasing the number of samples, but I was satisfied with this level of performance as a test case, especially because the rates used here were not realistic or reminiscent of real data, and the clusters were overlapping quite a bit.}

4.3.1.4 Over-specification of states and neurons

One issue that I have glossed over so far, is that in the previous two examples, we had given the \textit{true} number of states, as well as the \textit{true} number of hidden neurons to the model. This was done to facilitate a direct comparison between the underlying and the inferred parameter values. However, one may wonder how well our approach will work in practice, where we don’t know the true number of hidden states (in fact, there is no true number of hidden states for real data even), and we also don’t know the true number of hidden neurons.

Thankfully, it turns out that these parameters are not necessarily that important.

\begin{figure}
\centering
\includegraphics[width=0.3\textwidth]{feature.png}
\caption{Over specifying number of hidden neurons simply partition the feature space more finely. Even though there are } N = 3 \text{ true neurons, an over-specification of the number of hidden neurons will simply partition the feature space more finely.}
\end{figure}

As shown in Figure 4.5, an over-specification of the number of hidden neurons simply lead to a finer partition of the feature space (and hence more parameters to
estimate), but functionally such a repartitioning should have little to no effect on our model’s performance.

Figure 4.6: Over-specifying the number of hidden neurons has almost no effect on parameter estimation.
Top: rate maps (left: true, right: inferred). Middle: transition matrices (left: true, middle: direct estimation from true state sequence, right: inferred). Bottom: true state sequence (top), and decoded state sequence (bottom). Decoding accuracy is 97.5% on $T = 200$, as before.

Indeed, as shown in Figure 4.6, the model’s decoding accuracy was unchanged, and even the transition matrix estimation was unaffected. The rate maps can no longer be directly compared, but we see that the estimated rate map qualitatively lower firing rates, since the spikes are distributed across more neurons.

If, instead, we were to over-specify the number of hidden states, then the transition matrix will be slightly harder to interpret, but we again expect the model’s qualitative
performance to remain largely unchanged. Indeed, as shown in Figure 4.7, there are now twice as many states as true underlying states. However, we note that the states effectively now have finer resolution, and if we collapse all the blue-ish states into a blue super state, and similarly collapse all the red-ish states into a red super state, then the decoding accuracy is still at 97.5%, as before.

![Figure 4.7: Over-specifying the number of hidden states has almost no effect on decoding accuracy.](image)

Top: rate maps (left: true, right: inferred). Middle: transition matrices (left: true, middle: direct estimation from true state sequence, right: inferred). Bottom: true state sequence (top), and decoded state sequence (bottom). Decoding accuracy is 97.5% on $T = 200$ if all blue states are considered the same, and all red states are considered the same.

Even the inferred rate map is quite interpretable, where we see the same qualitative pattern as in the true rate map, but where each of those two states have each been split into two additional similar-but-more-nuanced sub states.
4.3.2 Experimental data

Even though the toy examples shown in section 4.3.1 showed relatively good performance, they were generated with idealized multivariate normal clusters, with truly fixed (homogeneous) firing rates, and so on. Consequently, we really wanted to see if our model performance would generalize to real, messy, multiprobe data.

The data were recorded from area CA1 in the hippocampus of a rat while the animal was traversing a 1 m linear track for liquid reward for a duration of about 15 minutes\(^2\). We had recordings from 6 tetrodes, and when manually clustering the data, we found 29 cells, of which about 9 were putative place cells.

![Figure 4.8: Feature space partitioning of a single probe from the linear track data. The feature space (peak amplitudes on each of the four tetrode channels) has been partitioned by our multivariate normal hidden neurons with diagonal covariance matrices.](image)

We have already shown that an over-specification of the number of hidden neurons should not pose any major challenges to our model (see section 4.3.1.4), and in our current implementation of our clusterless HMM, we have only implemented multivariate normal distributions with diagonal covariance matrices. There is no theoretical challenge to adding support for full covariance matrices however, and in fact, doing so will enable us to use more parsimonious representations of the clusters,

\(^2\)Data was recorded by J. Chu, from animal ‘install’.
since many real-world waveform features look like rotated Gaussians, as shown in Figure 4.8.

We fit our clusterless HMM with multiprobe multichannel data during run epochs on a linear track, using $Z = 30$ hidden states and $N = 30$ hidden neurons. a. The state transition matrix show the banded diagonal structure expected from traversals through the linear track. b. The lsPFs clearly show that states are well localized in space. c. Decoding to position using the lsPFs show a clear correspondence with the true underlying position. Note that decoding to the (spatial) mean of each state distribution was used, whereas we can get a closer correspondence if we decode to the peak location of each distribution.

We fixed our number of hidden neurons to $N = 30$, set the number of hidden states
to $Z = 30$, binned the data into 400 ms observation windows during run epochs$^3$, and fit our model using $M = 5000$ samples. Similar to our previous results, the model performance was qualitatively similar for a relatively large range of numbers of states (results not shown).

From Figure 4.9(a) we see that the inferred transition matrix again exhibits the characteristic banded diagonal structure—more precisely, the super diagonal appears to be the most dominant—consistent with movement along a linear track. We also find that the lsPFs localize very well in space (Figure 4.9(b)), and that we are able to decode to position (via the state space) reasonably well, too (Figure 4.9(c)). Note however that the decoded position shown in Figure 4.9(c) made use of the mean location for each state, which is why the decoded traces appear to be biased towards the center of the track. If instead we used the modes (peak locations) of the lsPFs, the correspondence would be even closer. Nevertheless, which method to decode (whether the peaks in a maximum likelihood sense, or the means in a maximum a posteriori sense) is a subject of some debate, and the choice should likely depend on your particular application or goal.

### 4.4 Related work

With the meteoric rise in popularity of machine learning and data mining, it is not surprising that researchers have tried to find ways to interpret neural data directly, without having to perform spike sorting first. The problem of spike sorting is made worse, in fact, by the equally rapid rise in the size of datasets. Instead of having to sort tens of cells, several researchers are now recording from 100s or even 1000s of cells, making it wholly impractical to sort spikes by hand. There seem to then only two reasonable solutions, namely (i) develop fully automatic and scalable spike sorting algorithms (see e.g., Chung et al., 2017), or (ii) develop algorithms to directly operate on the multiunit data, i.e., clusterless approaches. Here, we are chiefly concerned with the clusterless approaches.

As mentioned earlier in this chapter, one of the key motivations for going full clusterless is the high degree of subjectivity and variability in most spike sorting work-

---

$^3$The data quality was not the best, and we have previously found that we need about 500 ms bins to decode well with a supervised Bayesian decoding approach. Here, we have more data due to the inclusion of unsorted units, so that we decided to lower the observation window slightly for better temporal resolution.
Chapter 4  Clusterless hidden Markov models of neural activity

flows (Febinger et al., 2018). And indeed, clusterless decoding, being fully automated, is reproducible, and moreover has been shown to improve decoding performance (Ventura, 2008; Kloosterman et al., 2013), and allow for unbiased estimates of stimulation parameters (Ventura, 2008, 2009). There are cautionary tales in the space of clusterless decoding as well, however, such as the finding that including waveform features indiscriminately in a clusterless approach can contribute more noise and bias than useful information (Matano and Ventura, 2018). However, they continue to show that if features are chosen wisely and in a principled manner, that the decoding performance can be improved substantially.

Spike-sorting free approaches have been around for well over a decade (see e.g., Luczak and Narayanan, 2005; Ventura, 2008), but it has arguably become more prominent since the work by (Kloosterman et al., 2013), where they developed a nonparametric spike feature decoding approach using a time-homogeneous spatio-temporal Poisson process and density estimation to directly model the association between position and spike waveform features. Their approach was extended for real-time applications by using a kernel density compression algorithm by Sodkomkham et al. (2016). Indeed, a lot of recent research in clusterless approaches have focused on the real-time aspect of decoding (Sodkomkham et al., 2016; Deng et al., 2016; Yousefi et al., 2018; Arai et al., 2018; Michon et al., 2018).

Deng et al. (2015) extended the approach of Kloosterman et al. (2013) by developing an iterative marked point process filter; they similarly define a joint model for the spike waveform and receptive field (behavioral correlate) structure, but in addition, they augment this model with a state-space model to incorporate knowledge of the underlying signal they wish to decode. More recently, this same marked point process filter approach has been extended by Arai et al. (2018) to include an adaptive encode-decode model, where they suggest adaptively updating the model parameters during animal locomotion, and then fix the parameters and decode during bouts of quiescence (including PBEs). This is indeed an exciting / interesting framework, but it still presupposed that the neural activity during quiescence resembles that observed during active behavior. Indeed, to the best of my knowledge, there are no existing works on unsupervised clusterless approaches, where there is no observable behavioral correlate to model jointly with the spike waveform features.
4.5 Discussion

In this chapter we have introduced a new *unsupervised* clusterless model, the clusterless hidden Markov model. This contribution is novel in the sense that unsupervised clusterless approaches don’t seem to exist yet, as well as in the sense that, to the best of our knowledge, no work has been done on “binning” marked point processes as we’ve done here.

As an early effort, the model seems to perform well in terms of parameter estimation and decoding ability, but the model is still impractically slow. For the real data example presented in section 4.3.2, the model took about 24 hours to fit on a relatively modern desktop workstation. There are a number of ways in which we might attempt to improve this performance, including (i) parallelization of the sampling steps on Graphical Processing Units (GPUs), or by (ii) developing fast approximations to (4.7), or by (iii) foregoing the EM framework and instead focus on typically more scalable approaches such as VB, and finally by (iv) re-thinking a way to compute (4.7) directly without the need for sampling, in a similar fashion as what we’ve done with (4.14).

Nevertheless, the clusterless HMM presents an exciting opportunity to go back and re-analyze some of our previous work to see if the inclusion of more spike data will allow us to detect more / more robust instances of model congruence and / or replay events.
LATENT variable models have been used extensively in neuroscience, and the HMM in particular has found widespread use in areas ranging from finance, to speech recognition, to seismology (and of course in neuroscience, too). However, in most of those applications, the intent is often to improve decoding accuracy, by augmenting our observe-decode model with latent factors that help us relate the observations to the desired decoded outputs.

Somewhat unusual then, we have used the HMM in an exploratory data analysis mode instead, by building models of PBEs and asking directly what we can learn about the PBEs from those models. We found that our PBE-first approach resulted in models which still captured the spatial structure of the behavioral tasks we studied. This is interesting for a number of reasons. First, only a small number of PBEs are usually classified as “significant” (meaning spatially-consistent) with the surrounding animal behavior, so it was unclear that we would be able to find any meaningful structure from so little (and so noisy) data. Second, it suggests that PBEs could perhaps provide all the necessary information about experiences to downstream regions in the brain.

It has been previously observed that the rate of hippocampal reactivations in PBEs during awake behavior is much higher than during sleep (Grosmark and Buzsáki, 2016;
Karlsson and Frank, 2008), but the reasons for this are not well understood. One hypothesis is that many sleep PBEs contain the reactivation of contexts other than those measured during a behavioral experiment. Another hypothesis is that sleep activity involves remodeling of dynamic network architectures (Buhry et al., 2011; Tononi and Cirelli, 2014). Our approach has the potential to illuminate some sources of variability during sleep. While we have given preliminary evidence that information about a remote context can be present in PBEs along with the local context, further work is required to understand how our model’s ability to capture this structure scales with the number of different contexts. With sufficient data, our HMM approach should be able to learn disjoint sets of latent states (or “sub-models”) which would capture these separate contexts and allow us to test this possibility. Alternatively, sleep PBEs could yield models which represent a known behavioral context but are markedly different (e.g., less sparse) than those learned from awake PBEs. This might support the network remodeling function of sleep. In the latter case, we might imagine that only a small subset of sleep PBEs—corresponding to learning-related replay—would be congruent with a model learned from awake PBE data.

Our clusterless HMM holds great promise in allowing us to perform the above analyses using more data than would be possible in the sorted unit case, but realistically it needs to be made significantly faster before it can really be considered practical. Fortunately, there are several promising ways to speed up our implementation, as discussed in section 4.5.

Overall, we believe that we have demonstrated the usefulness and advantages of unsupervised approaches to study internally-generated neural activity, and with our new clusterless HMM, we believe that we will soon be able to analyze an even broader class of neural activity, and in particular those short and sparse events that were previously marked with having too few spikes to analyze in any meaningful way.
Bibliography


Bibliography


Bibliography

Chen, Z., Kloosterman, F., Brown, E. N., and Wilson, M. A. (2013). Uncovering spatial topology represented by rat hippocampal population neuronal codes. The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters. Citation Springer Science + Business Media B. V. (Cited on page 12.)


Bibliography


LIST OF ABBREVIATIONS

PBE  Population Burst Event
SWR  Sharp Wave Ripple
AUC  Area under the curve
ROC  Receiver Operating Characteristic
EM   Expectation Maximization
VB   Variational Bayes
CA   Cornu Ammonis
LFP  Local Field Potential
IsPF Latent State Place Field
HMM  Hidden Markov Model
SWS  Slow Wave Sleep
REM  Rapid Eye Movement
GPU  Graphical Processing Unit