Assessing neurocognitive hypotheses in a likelihood-based model of the free-recall task*

Sean M. Polyn

July 2022

Introduction

In the free-recall task, a participant studies a list of words, usually presented one at a time. This is followed by a recall period in which the participant reports as many items as they can from the study list, in whatever order they come to mind. As such, behavioral responses in this task are characterized as recall sequences: The series of responses made by the participant during the recall period. Theorists have developed a variety of cognitive models designed to account for the behavioral dynamics observed in different versions of the free-recall task. These models often describe a hypothetical set of cognitive operations that give rise to the observed behavioral data.

This chapter provides a tutorial overview of a particular application of free-recall modeling, in which we assess neural signals in terms of their potential correspondence with the cognitive processes embodied in a model of free recall. Specifically, we examine a set of analyses described by Kragel et al. (2015), in which blood-oxygen level dependent (BOLD) signals recorded with fMRI were related to mechanisms from a retrieved-context model, using a direct-input approach, in which neural signals are used to directly control aspects of a behavioral model (Turner et al., 2017; Purcell et al., 2010). The reader is invited to download companion Python code that implements many of the simulations described in this chapter, hosted at `github.com/vucml/ncms_toolbox`. The README file at that URL will direct the reader to the Python script containing the tutorial (`KragEtal15_tutorial.py`) as well as directions for installing the `cymr` toolbox the tutorial uses

*This work was supported by a grant from the National Science Foundation to SMP (#1756417). Cite as: Polyn, S. M. (2022) Assessing neurocognitive hypotheses in a likelihood-based model of the free-recall task. In Forstmann, B. U. and Turner, B. (Ed.). Introduction to Model-Based Cognitive Neuroscience (2^{nd} edition). Springer, New York.

to run the simulations.

Many models of free recall have been proposed and examined over the decades, such as the Search of Associative Memory model (SAM; Raaijmakers & Shiffrin, 1981), the Temporal Context Model (TCM; Howard & Kahana, 2002), and the Scale Invariant Memory, Perception, and Learning model (SIMPLE; Brown et al., 2007), among others (Davelaar et al., 2005; Lehman, M. and Malmberg, K. J, 2013; Farrell, 2012). The examples developed in this chapter draw heavily from my own work with retrieved-context models of free recall (which include TCM). The family of retrieved-context models contains many variants with similar names, certain unique properties, and a strong family resemblance (Howard et al., 2005; Sederberg et al., 2008; Lohnas et al., 2015; Healey & Kahana, 2014). Here, we will examine the Context Maintenance and Retrieval model (CMR; Polyn et al., 2009), a retrieved-context model examined by Kragel et al. (2015), but some other variants will be discussed as well. Before getting to the tutorial, I briefly review the structure and operation of CMR, highlighting mechanisms and processes that figure prominently ahead.

Overview of the Context Maintenance and Retrieval (CMR) model

A named model (e.g., CMR) is often a moving target. The particulars of its implementation can change from paper to paper, and often different versions of a model are examined in a single paper. As such, it can be useful to think of CMR (or TCM) as a *modeling framework*, rather than as a singular *model*. This phrase emphasizes that in any given paper, different versions of the model will be constructed, assessed, and compared to one another.

For example, in our paper introducing CMR (Polyn et al., 2009), we were interested in characterizing how shifting from one encoding task to another during study affected recall performance. We constructed three model variants in which task shifts had different effects on the operation of the model. The neural application described below (from Kragel et al., 2015) used a version of CMR with the same basic structure, but task information wasn't explicitly simulated. Here, we review the Kragel et al. version of CMR, which we used to characterize the relationship of neural signals from the medial temporal lobe to the cognitive processes defined by the model.

CMR is a cognitive process model. It is an assembly of mechanisms and processes, and a set of rules determining the sequence in which they are engaged, and the model defines the consequences of that engagement. Following the notation of Turner et al. (2017), the parameters θ control these model mechanisms, allowing it to make predictions about the observed behavioral data B. A vector

of parameter values θ, each with an allowable range, defines a *parameter space*, where a given point in the space corresponds to a particular configuration of the parameters.

Optimization of the model for a particular set of behavioral data (B) involves searching through this parameter space to find the specific parameter set that allows the model to best predict, or fit, the data. This can be done by using the model to generate a large number of synthetic recall sequences, calculating a set of summary statistics (e.g., serial position curve, and lag-based conditional response probability curves), and evaluating how closely those summary statistics match the results of the same analyses on the observed data (B). Examples of this approach can be found in Sederberg et al. (2008) and Polyn et al. (2009). Here, we used an alternative approach to optimization, in which the model's goodness of fit is quantified in terms of the likelihood of the specific sequences of recall events that comprise B. This allowed us to incorporate trial-specific neural signals into the model, as will be described below. Examples using this type of likelihood-based optimization approach with a free-recall task can be found in Kragel et al. (2015) and Morton & Polyn (2016).

Basic operation of the model

The CMR model is a simplified neural network with two representational layers, each implemented as a vector space. The model is depicted schematically in Figure 1b. The **F** layer represents the features of particular items as they are studied or remembered. Activation of the i^{th} study item representation is indicated by the vector $\mathbf{f}_i$. Each of these item representations is a unit vector, with one element of the vector set to 1 and the other elements set to 0. The **C** layer contains a gradually changing representation of temporal context, with specific states of the context layer indicated by the vector $\mathbf{c}_i$.

CMR is a linear associative network (Anderson et al., 1977) in which the two layers influence one another via two associative weight matrices. The $\mathbf{M^{FC}}$ weight matrix projects from the feature layer to the context layer, and $\mathbf{M^{CF}}$ contains the recurrent connections from the context layer back to the feature layer. Each matrix can be thought of as having two sets of associations, a set of pre-experimental associations built into the network when it is initialized ($\mathbf{M^{FC}_{pre}}$ and $\mathbf{M^{CF}_{pre}}$), and a set of experimental associations learned over the course of a trial ($\mathbf{M^{FC}_{exp}}$ and $\mathbf{M^{CF}_{exp}}$). These two components combine additively, i.e., $\mathbf{M^{FC}} = \mathbf{M^{FC}_{pre}} + \mathbf{M^{FC}_{exp}}$.

When the model is initialized, the pre-experimental associations allow a study item to retrieve information about the past contexts it has been experienced in (also known as the item's *pre-*

experimental context). So, when an item representation $\mathbf{f}_i$ is activated, a matrix multiplication operation projects the item representation through these associative connections to determine the net input to the context layer:

$$\mathbf{c}^{IN} = \mathbf{M}^{\mathbf{FC}}\mathbf{f}_i \tag{1}$$

The units in the context layer have a special integrative property. Incoming activation ($\mathbf{c}^{IN}$) only partially displaces the previous activation state ($\mathbf{c}_{i-1}$):

$$\mathbf{c}_i = \rho\mathbf{c}_{i-1} + \beta\mathbf{c}^{IN} \tag{2}$$

Here, β is a scalar parameter that weights the influence of $\mathbf{c}^{IN}$. The value of the ρ parameter is then calculated to ensure the magnitude of $\mathbf{c}_i$ is constant (at unit length). As new items are presented, new context units are activated, and activity in the other units fades away exponentially. As such, the state of context at a given point in time is a recency-weighted average of the model's past experience.

The second associative matrix, $\mathbf{M}^{\mathbf{CF}}$, connects the context layer back to the item layer, making the model a kind of simple recurrent network (SRN). The relationship of the temporal context model to other SRNs is discussed by Howard & Kahana (2002). We will return to the role of this second matrix momentarily, as it is more influential during the recall process.

When an item is studied, both associative matrices are updated using a Hebbian learning rule that links the item representation to the current contextual representation (the item's *experimental context*). These experimental associations ($\mathbf{M}^{\mathbf{FC}}_{\mathbf{exp}}$ and $\mathbf{M}^{\mathbf{CF}}_{\mathbf{exp}}$) embody the set of episodic memories formed during the study period. Neighboring items on the study list are associated with similar states of context, and for any two study items, the similarity of their corresponding context states decreases as the spacing between the items increases.

The experimental associations imbue the network with two special powers: They allow the context representation to act as a cue to prompt the retrieval of study items, and they allow an item representation to prompt the retrieval of contextual states from the study period. Starting with the contextual cueing process:

$$\mathbf{f}^{IN} = \mathbf{M}^{\mathbf{CF}}\mathbf{c}_i \tag{3}$$

In words, the context representation projects through the associative matrix to activate a blend of item features, $\mathbf{f}^{IN}$. This sets in motion a retrieval competition, where the likelihood of a given item being recalled is proportional to how strongly it is activated in the $\mathbf{f}^{IN}$ blend. When an item wins this retrieval competition, its features are reactivated on the item layer. Now the item representation can be used to reactivate the past state of context linked to that item during the study period, a process called *temporal reinstatement*. The dynamics of temporal reinstatement are given in Equations 1 & 2. The same process that updates context during the study period also updates context during memory search. The β parameter governing the degree of contextual updating is allowed to take different values during the study/encoding period (β_{enc}) and during the search/recall period (β_{rec}).

If β_{rec} is zero, there is no temporal reinstatement during recall; the temporal context representation is unaffected by the contextual information associated with the remembered item. If β_{rec} is one, there is complete integration; the temporal context representation is completely overwritten by the contextual information associated with the studied item, i.e., the temporal context of the study event is perfectly reinstated. Between these two extremes, temporal reinstatement partially overwrites the current state of context. After the recall event, context is a blend of the prior contextual state and the retrieved context, with larger values of β_{rec} indicating more successful temporal reinstatement.

It is this process of temporal reinstatement that gives rise to the behavioral phenomenon of temporal organization, whereby items that were studied in neighboring list positions tend to be recalled successively. When a past state of context is reinstated, it becomes part of the retrieval cue guiding the next retrieval competition. Because the context representation changes gradually during the study period, this reactivated context is a good cue for items from neighboring list positions, giving them a preferential boost in the retrieval competition. In the neural simulations described below, we allow neural signal recorded from the medial temporal lobe to control the temporal reinstatement process (Figure 1). Turner et al. (2017) refer to this as a *direct-input approach*. We estimated neural parameters δ which were used to control the behavioral parameter β_{rec}, allowing us to evaluate whether this neural influence improved the ability of the model to predict the behavioral data B.

The final relevant component of CMR dynamics is a process that determines when recall terminates. Each recall competition is governed by a probabilistic decision rule (Luce, 1986; Howard & Kahana, 2002) where the activation of each item representation in $\mathbf{f}^{IN}$ determines its likelihood

of winning the recall competition. However, there is also the possibility that none of the items will be retrieved, meaning the recall process has terminated.

We use an equation that captures the steady growth in the likelihood of recall termination as a function of output position (Dougherty & Harbison, 2007; Miller et al., 2012). The growth rate of the recall termination likelihood is an adjustable parameter of the model. A lower growth rate means the model will enjoy more recall successes (on average) before the search process eventually terminates. In the neural application developed by Kragel et al. (2015), we allowed neural signals in medial temporal lobe to either influence the temporal reinstatement process (as mentioned above), or this recall success process. This allowed us to examine whether a given candidate signal indicated a generic boost in the likelihood of recall (recall success), or a specific increase in the likelihood of a neighboring item (temporal reinstatement). In the tutorial simulations described below, we focus on the temporal reinstatement process.

Evaluating the model

There are two useful ways to evaluate the performance of the CMR model. For a given parameter set θ, one can generate synthetic recall sequences, or one can take a set of observed recall sequences and determine the likelihood they were produced by the model. Polyn et al. (2009), for example, examined different variants of CMR. Each variant was used to generate a large number of synthetic recall sequences. Summary statistics were calculated for both the observed data and these synthetic recall sequences, and the goodness of fit of a given parameter set was a function of how well the observed and synthetic statistics matched one another (using, e.g., a chi-squared statistic or root mean squared deviation).

The summary statistics used by Polyn et al. (2009) included serial position curves, probability of first recall curves, and lag-CRP curves (which characterize temporal organization). In all, 93 different behavioral data points were used, across these and other measures. A chi-squared statistic was used to determine goodness of fit. This statistic has the nice property of normalizing each data point in terms of the standard error of the underlying behavioral measure. This approach allows the modeler to specify which aspects of the data will be most influential in determining the best-fitting parameters. As such, it requires a number of decisions to be made regarding the relative importance of different behavioral measures. With this approach, the identity of the optimal set of parameters can change depending on which summary statistics are used to calculate goodness of fit.

Kragel et al. (2015) used the likelihood-based approach, in which a parameter set is evaluated in terms of the model's ability to predict the specific recall sequences observed in an experiment. The model assigns probabilities to events, and the fitness of the model is dependent on how well those assigned probabilities match the actual probabilities of those events in the observed data. To use this approach in free recall, we treat the recall sequence on a given trial as a series of discrete responses: recall events. For each recall event, we calculate the probability of each possible response. This allows us to assign a probability to each recall event: The likelihood that the model would have produced the observed response made by the participant. We can then determine the likelihood of the model producing the full sequence of responses on a given trial, and by extension the likelihood of the model producing the data set as a whole (for a given parameter set).

Let's look closer at the behavioral data collected in a standard free recall task. If the task uses spoken recall, each trial has an associated audio recording of the participant's verbal responses. An annotator (human and/or machine) marks the identity and onset time of each word that's reported by the participant. For now, we can imagine that we've done some simplifying steps, such as excluding intrusions (any reported words that weren't actually on the target study list) and repetitions. In certain applications we attempt to simulate the timing of the individual responses, but here we simply focus on the sequence without regard to the timing.

Each valid response is labeled with an integer corresponding to the remembered item's position on the study list. These are the building blocks of the recall sequences. For a single trial, the length of the sequence can be anywhere from 0 (if no valid responses are made) to the length of the study list (if every studied item is successfully recalled). If we include repeated responses and intrusions, the length of the sequence and the set of possible responses are not as well constrained. If intrusions are allowed, all words in an individual's lexicon are technically possible responses. Even with repeats and intrusions, a predictive approach is certainly possible, but requires extra work to allow the model to specify the likelihood of these other responses.

Note that this model isn't a stationary process over the course of the recall period. According to the basic theory described above, each recalled item alters the composition of the contextual retrieval cue. If the first response is from the second list position, the model's predictions regarding the second recall event will be very different than if the first response is from the 15^{th} list position. Each recall event alters the set of retrieval probabilities assigned to the not-yet-recalled items.

The optimization process, instead of optimizing the match between observed and synthetic summary statistics, attempts to maximize the likelihood of observing a given data set for a given

model. What can we say about this probability space? We start by considering an individual trial in more detail. Even with the restrictions outlined above (exclusion of repeats and intrusions), the set of possible outcomes on a given trial can be absurdly large, but it is not infinite. Given that we have excluded repetitions, a sampling without replacement process can describe the set of possible recall sequences. How many possible recall sequences are there for a given list length? Equation 4 describes the necessary calculation to determine this. The set of possible recall sequences is equal to the set of distinct ordered subsequences (i.e., permutations) of all possible sequence lengths from 0 up to the list length. We count length zero as a possible outcome, as it represents the scenario where a person fails to recall any valid items on a given trial. This is a rare event in practice, but it does happen.

$$N_{seq} = \sum_{r=0}^{K} |P(K, r)| \tag{4}$$

In Equation 4, K indicates list length, r is the length of a particular recall sequence, $P(K, r)$ specifies the set of possible permutations when r items are chosen from a set of K items, and the vertical bars surrounding this indicate the cardinality of this set (i.e., the number of elements in the set). With $K = 1, N_{seq} = 2$ (nothing is recalled, or the one studied item is recalled). With $K = 3, N_{seq} = 16$. With $K = 8, N_{seq} = 109,601$, and a list length of 24 (a reasonable length for a free-recall task) yields something in the neighborhood of 1.7×10^{24} possible recall sequences.

We engage in this exercise not to despair at the possibility of considering each of these possible outcomes, but rather to set expectations regarding the scale of the likelihood values that will be produced by a computational model. Given these myriad possibilities, the likelihood of observing a specific recall sequence can be minuscule. In practice, we have no need to enumerate all possible outcomes of an experiment, we just need to calculate the likelihood of observing a particular outcome, given a particular model. Then we can compare the likelihood of that outcome under different (possibly nested) variants of the model, to evaluate which model would be most likely to produce the observed recall sequences. It is ok that particular recall sequences are assigned very low probabilities; the important thing is the relative likelihood of the recall sequences under different models. In other words, recall sequence X could have a one-in-ten-thousand likelihood under model A, but a one-in-a-million likelihood under model B. If similar trends hold up across many recall sequences, model A will be preferred.

The above exercise demonstrates that likelihood values at the trial level (the probability of

observing a particular recall sequence) will be very small. Following standard convention, we log-transform these probability values to log-probabilities. This allows us to avoid potential problems while running our simulations on digital computers, e.g., where an exceedingly small floating point number can be rounded off to zero. The difference between an exceedingly small number and zero is of great practical importance when dealing with probabilities, because zero indicates an event is impossible. If a model assigns any observed event a zero probability, then from the point of view of model comparison, it is impossible that the model gave rise to the observed data.

Calculating the likelihood of a given recall sequence under a given model is fairly straightforward. As described above, each trial has an associated recall sequence, which is a series of recall events. Each recall event can be represented by a categorical distribution (sometimes called a generalized Bernoulli distribution) where each potential outcome has an associated probability. For a list of length L, it is useful to define $L + 1$ possible outcomes for a given event: One for the potential recall of each study item, and one to represent termination of the recall sequence. The probability associated with each of these outcomes is determined by the model, given a particular set of parameters. Equation 5 calculates the probability of a given recall sequence (p_{seq}), where p_i is the probability assigned to the i^{th} recall event. The likelihood of the entire sequence is simply the product of the probabilities of the individual recall events.

$$p_{seq} = \prod_{i=1}^{r+1} p_i \tag{5}$$

$$L_{seq} = \sum_{i=1}^{r+1} L_i \tag{6}$$

Probability models such as the binomial distribution and multinomial distribution take a somewhat different form than Equation 5. These other distributions model the probability of counts of a particular outcome across a certain number of trials. In contrast, free recall is better described using a sequential sampling process. Specifically, sequential sampling of a finite population without replacement (Mallows, 1973). Equation 6 shows the same calculation as Eq. 5, but on the log-transformed probabilities of the individual recall events (L_i). In this case, the log-probability of the recall sequence is simply the sum of the L_i values assigned to each event in the sequence. To be clear, the p_{seq} and the L_{seq} values refer to the same thing: $log(p_{seq}) = L_{seq}$, and $e^{L_{seq}} = p_{seq}$. The probability of an entire experiment is calculated as either the product of all the trial-level probabilities, or the sum of all the trial-level log-probabilities (usually referred to as *log-likelihood*).

The aggregate log-likelihood number is not meaningful in and of itself; it is a sum over recall events, so simply adding more trials to a data set will cause the log-likelihood to become more negative. However, if two models are applied to the same data set, their corresponding log-likelihood scores can be meaningfully compared.

Regardless of whether one evaluates a model using summary statistics or event likelihoods, similar parameter optimization techniques can be used. Using the first approach described above, an optimization algorithm (e.g., a particle swarm) is used to find the set of parameters that allow the model to produce synthetic data whose summary statistics match the observed summary statistics in a given experiment. Using the event likelihood approach, the same optimization algorithm can be used, but now the goal is to find the set of parameters that maximizes L_{seq}, the log-likelihood of the data given the model. Generally speaking, log-likelihood scores will be negative values, with values closer to 0 indicating that the model is making more accurate predictions. If the model makes perfect predictions (it won't), always assigning a probability of 1 to each observed response, the log-likelihood will be 0. However, many common optimization routines are designed to minimize, rather than maximize, whatever function is fed into it. In these cases we simply multiply the log-likelihoods by -1, and carry on with optimization.

One can then perform model comparison on a set of candidate model variants being compared with one another. Usually, parameter optimization is carried out for each model variant, yielding a best-fit log-likelihood for each model. The one with the maximal log-likelihood (i.e., closest to 0) is the one that is most consistent with the observed data. There are a variety of model comparison statistics that help us characterize whether differences in log-likelihood between models are reliable and worth further consideration. We will return to model comparison techniques below.

Assessing a neurocognitive linking hypothesis

Here, I provide a tutorial overview of an application of event-likelihood modeling of free recall, in which we incorporated neural signals into CMR to assess a set of neural linking hypotheses (Kragel et al., 2015). Code is provided for you to explore a simplified version of the analyses presented in that paper. First, I will review the questions of interest, and the general technique. The neural linking hypotheses examined in this study link a particular neural signal with a cognitive process defined by the CMR model. The neural signals of interest are blood oxygen level dependent (BOLD) signals in medial temporal lobe (MTL) structures recorded with functional MRI. These signals can

be thought of as neural parameters δ estimated from the fMRI data using a general linear modeling approach. These neural parameters δ are linked to the temporal reinstatement mechanism in the model described above.

Neural circuitry in the MTL is thought to be critically involved in the formation and retrieval of episodic memories. This idea is supported by neuropsychological studies demonstrating that damage to MTL structures devastates a person's ability to form new episodic memories (Milner et al., 1998). The Complementary Learning Systems model of McClelland et al. (1995) suggests that the hippocampus (a brain structure within the MTL) plays a key role in the formation of new episodic memories by allowing the rapid formation of associations linking the myriad details of a particular experienced event to one another, and to the broader spatiotemporal context in which the event occurs (Norman & O'Reilly, 2003; Schapiro et al., 2017). Endel Tulving, in his writings on the episodic memory system, described the phenomenon of mental time travel, whereby an individual can reactivate the contextual details of a past experience with enough vividity that it is like they are revisiting the past experience (Tulving, 1993). To the extent that this kind of reminiscence relies on the recovery and reactivation of the spatiotemporal context of an event, we expect that successful mental time travel requires an intact hippocampal system.

Our goal in the Kragel et al. (2015) study was to determine whether we could relate moment-to-moment changes in blood flow in MTL regions to the behavioral performance of participants, thus allowing us to refine our understanding of how different subregions of MTL support memory-guided task performance. We used the CMR modeling framework to develop several model variants instantiating different neural linking hypotheses. In this context, a neural linking hypothesis is a proposal that links a particular neural signal to a particular cognitive operation in the model. Models with an embedded neural linking hypotheses are referred to as *neurally informed*, in that the strength of the neural signal influences the degree of engagement of the linked cognitive operation. Thus, fluctuations in the neural signal change the dynamics and therefore the predictions of the model, relative to a baseline, or *neurally naive* version of the model that isn't influenced by neural signal. To the extent that a neurally informed model does a better job predicting the participant's responses compared to a corresponding neurally naive model, the embedded neural linking hypothesis is supported, and deemed worthy of further examination. As described above, retrieved-context models describe a set of candidate cognitive mechanisms that support mental time travel. When the model retrieves the details of a specific past event, this prompts the system to retrieve associated contextual information, which then supports the retrieval of other events

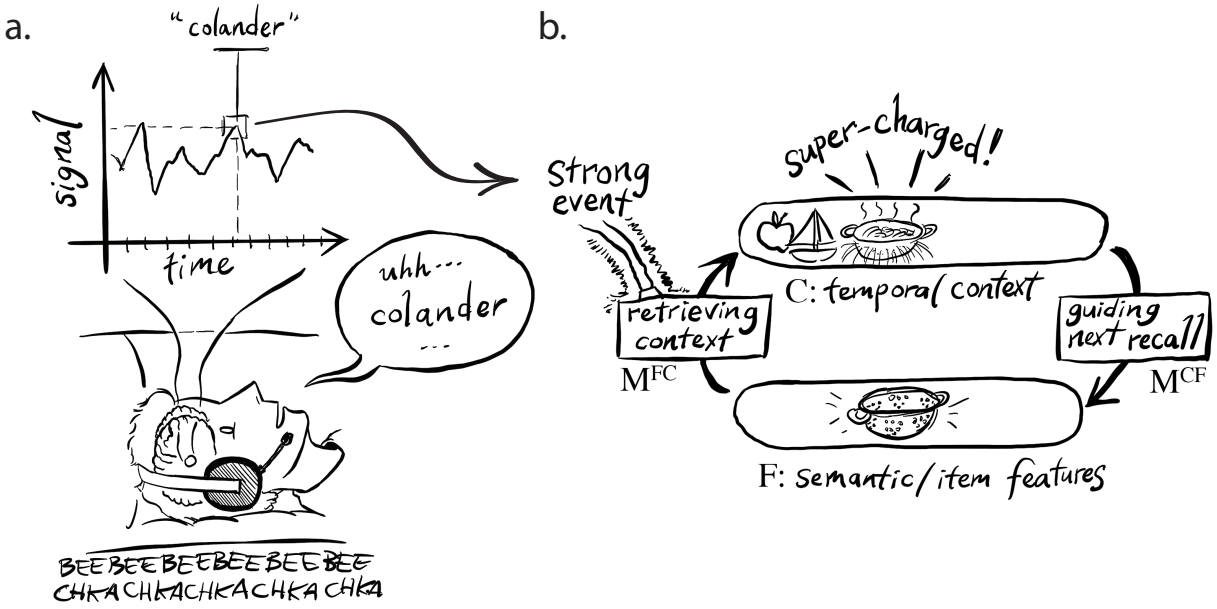

Figure 1: (a) A participant performs a free recall task while lying in an fMRI scanner. A microphone records their vocal recall responses. Brain signal is sampled at the time of each recall event. The signal strength is used to control the temporal reinstatement mechanism in the model. (b) When an item (e.g., colander) is recalled, the semantic representation of the item is activated in the F layer. This representation is projected along the $\mathbf{M^{FC}}$ associative connections to retrieve contextual information associated with the item. Here, the strong neural signal causes context information to be retrieved with high fidelity. Specifically, the strong neural signal increases the value of the β_{rec} parameter (see Equations 2 and 7). This will increase the likelihood that the next recalled item will come from an adjacent serial position (as depicted in Figure 2).

that occurred nearby in time. Thus, if we have a neural signal whose engagement indicates the successful reinstatement of temporal context, we can use the model to make detailed predictions about the behavioral consequences of that reinstatement.

Kragel et al. (2015) constructed a set of neurally informed *temporal reinstatement* models, to examine whether signal in medial temporal lobe was plausibly related to the temporal reinstatement process engaged during memory search. This is depicted schematically in Figure 1. One set of analyses examined each gray-matter voxel within MTL in turn. First, signal within that voxel was estimated at the time of each recall event. These neural response values (δ_{event}) were z-score normalized by trial. The following equation describes how these neural response values were used to update a linked model parameter:

$$\theta_{event} = \theta + \nu\delta_{event} \tag{7}$$

For the temporal reinstatement models, θ corresponds to the base value of the β_{rec} parameter. For each recall event, the neural scaling parameter ν is multiplied by the voxel's neural response, and is then added to the base value of β_{rec}, yielding the event-specific parameter value, θ_{event}. As such, fluctuations in the neural signal cause the β_{rec} parameter appearing in Eq. 2 to fluctuate. The neurally naive version of the model is realized by setting the ν parameter to zero. In this nested model, fluctuations in neural signal no longer influence the target parameter.

The model predicts that successful temporal reinstatement leads to temporal organization in the observed recall sequences. Figure 2 demonstrates how a behavioral measure of temporal organization, a lag-based conditional response probability analysis, or lag-CRP, is affected by the degree of success of the temporal reinstatement process. If temporal reinstatement is strong when a particular item is retrieved, the contextual state associated with that item's study event is strongly reactivated. This contextual state is a good retrieval cue for items from neighboring positions on the study list, so the next recalled item is likely to come from a nearby list position. Conversely, if temporal reinstatement is weak, there will be less of an advantage for the just-recalled item's neighbors.

Kragel et al. (2015) used the CMR model to evaluate whether fluctuations in a given neural signal during the recall period correspond to fluctuations in the degree of success of this temporal reinstatement operation. Equation 7 provides the interface between the neural signal and the computational model. When estimates of a given voxel's activity are used to populate the δ_{event}

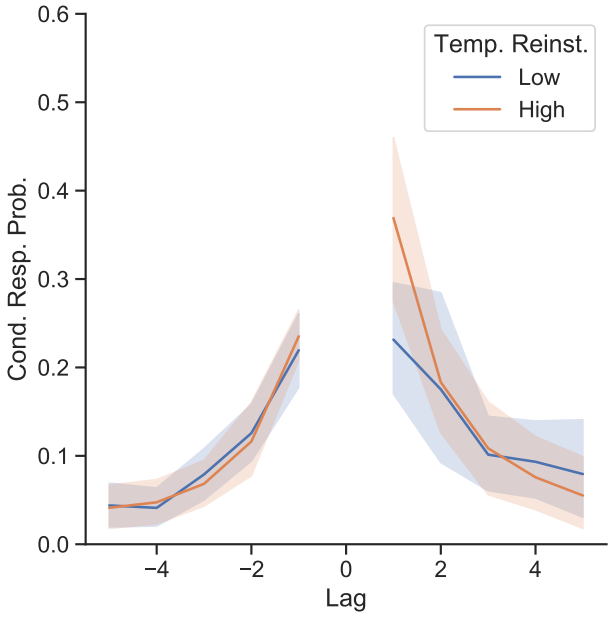

Figure 2: A lag-based conditional response probability analysis of synthetic recall sequences. This analysis calculates the probability of making recall transitions of particular lag distances, measured in terms of the items' serial positions on the study list. These two lag-CRP curves indicate how the temporal reinstatement parameter β_{rec} affects temporal organization. When β_{rec} is high (greater than 0.5, red triangles), there is an increased likelihood of nearby transitions relative to when β_{rec} is low (less than 0.5, black circles). For this parameter set, this exhibits as an increased likelihood of +1 recall transitions (recalling the next item from the study list).

matrix, fluctuations in voxel activity from recall to recall will influence the degree of temporal reinstatement, and through that, influence the model's predictions regarding the most likely behavioral response (i.e., the likelihood of recalling a particular item next).

The likelihood-based optimization procedure was used to find the parameter set that maximized the model's ability to predict each participant's recall sequences. The neural scaling parameter ν was one of the free parameters being estimated. If the best-fitting value of ν was positive, this meant the sampled neural signal was useful and informative for the temporal reinstatement mechanism, and improved the model's ability to predict the specific sequence of items that would be recalled. Specifically, with a positive value of ν, an increase in blood flow at that anatomical location increases the value of β_{rec} associated with that recall event.

To the extent that these fluctuations in blood flow actually correspond to fluctuations in the likelihood of contiguous recalls, the neurally informed model will make more accurate predictions, and end up with a better likelihood score. The likelihood score associated with the best-fitting neurally informed model was compared to the likelihood score of the best-fitting neurally naive model using a likelihood ratio test (Wilks, 1938), to determine whether any improvement in predictive power of the neurally informed model was statistically significant. [1]

A second set of neurally informed models were created in which a given voxel's activity was associated with the *recall success* mechanism described earlier. This allowed us to test whether signal in a given voxel indicated that more recalls were likely to be made, but without making the specific prediction that they would come from a nearby list position. We won't go into detail regarding the recall success mechanism here. The interested reader is referred to the Kragel et al. (2015) report, where this model variant is described in more detail.

Using this approach, the cognitive model becomes part of a neuro-behavioral statistical framework for interpreting the functional properties of neural activity. We constructed and tested neurally informed models (the temporal reinstatement and recall success variants) for each voxel in the MTL. This allowed us to make a map indicating which voxels contained signal that was informative for each model process. In other words, we were able to visualize the anatomical distribution of voxels whose activity showed a functional correspondence to these model-defined cognitive processes.

One of the central results of the paper was the identification of a functional gradient across the

[1] This model comparison may not have been strictly necessary: The neurally naive model is nested within the neurally informed model (in that the neurally informed model becomes identical to the naive model when $\nu = 0$). As such, a statistical technique demonstrating that the best-fit value of ν is reliably above zero would allow us to draw similar conclusions.

anterior-posterior axis of the MTL. More anterior voxels in medial temporal lobe cortex (MTLC) were more informative to the retrieval success model, indicating an involvement in memory retrieval, but not necessarily in temporal reinstatement. More posterior voxels in MTLC, and posterior voxels along the hippocampal axis, were more informative to the temporal reinstatement model, potentially indicating their involvement in context-guided memory search. The theoretical implications of these results are discussed further by Kragel et al. (2015).

Simulation exercises

In this section of the chapter we take a closer look at the CMR model, and present simulations designed to familiarize you with the prediction of recall sequences, and the generation of synthetic recall sequences. The URL for the companion code can be found in the Introduction section above. The code is divided into numbered sections that we will refer to in the text.

Exercise 1: Basic parameter recovery

In section one of the tutorial code, we create a data structure that specifies a number of model parameters. These parameters are set to reasonable values that allow the model to produce recall sequences generally consistent with the results of a standard immediate free-recall experiment (e.g., Kahana, 2012). After you've worked your way through the tutorial, you may wish to try changing these parameters to alter the dynamics of the model. It is certainly possible to pick parameter values that cause the model to perform poorly (e.g., never making a successful recall), or that will cause the code to execute improperly (e.g., if the β parameters are set outside of the range of 0–1). While you are getting your bearings, try changing parameters one at a time. In this section we also create variables that specify certain task parameters such as list length (set to 24) and the total number of trials, which is set to 120. These values were chosen to match the methodological details of the Kragel et al. (2015) experiment and simulations. As with the model parameters, these task parameters can also be altered to explore how changes affect the model's performance. The code in section 1 uses the task parameters to create a `pandas` data structure containing synthetic study events. This data structure contains a row for each study event, specifying (among other things) the participant's identity and the serial position of the presented item.

In section two of the tutorial code, we generate synthetic behavioral data using these model parameters. The generative function takes the parameter structure and the study event data structure

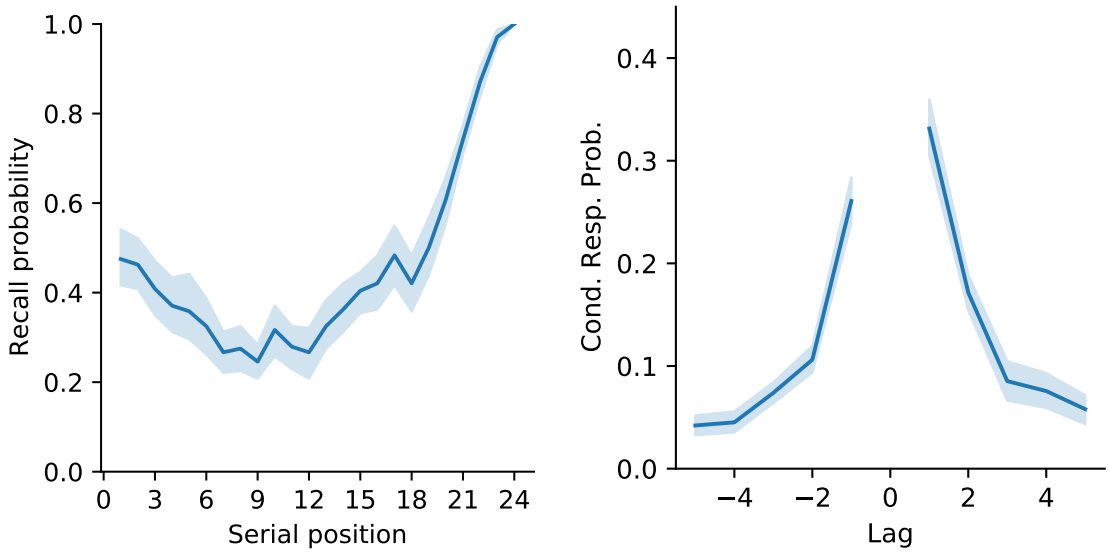

Figure 3: Examining recall performance from Exercise 1 of the tutorial code. (a) A serial position curve analysis of the synthetic recall sequences produced by the model. This calculates the probability of recalling each study item given its list position. (b) A lag-based conditional response probability (CRP) analysis on the synthetic recall sequences. This calculates the probability of successively recalling two items with a particular lag distance separating them. For example, recalling item 10 followed by item 11 is a recall transition of lag +1.

as input arguments. It returns a recalls matrix containing 120 trials worth of model-generated synthetic recall sequences. Two example summary statistics are calculated for these recall sequences: The first is a serial position analysis which calculates the probability of recalling particular items based on their serial position in the study list. The second is a lag-based conditional response probability analysis, which calculates the probability of making recall transitions of particular lag distances. For example, if a participant recalls the item from the fifth serial position followed by the item from the sixth serial position, this is a transition of lag +1. If item 6 was followed by item 4, this is a transition of lag -2. Detailed explorations of these and other common free recall analyses can be found in Kahana (2012). Figure 3 presents these two analyses for a sample run of the generative model.

Section 3 of the tutorial code runs a series of predictive simulations using the synthetic recall data from section 2. This part of the code provides a simple demonstration of parameter recovery. Parameter recovery generally refers to an attempt to determine the best-fitting parameters for synthetic data (i.e., a situation where the generating parameters are actually known). By evaluating the likelihood of a variety of parameter sets, one can determine whether the 'recovered' best-fitting parameters match the parameters used to actually generate the synthetic data. This process helps

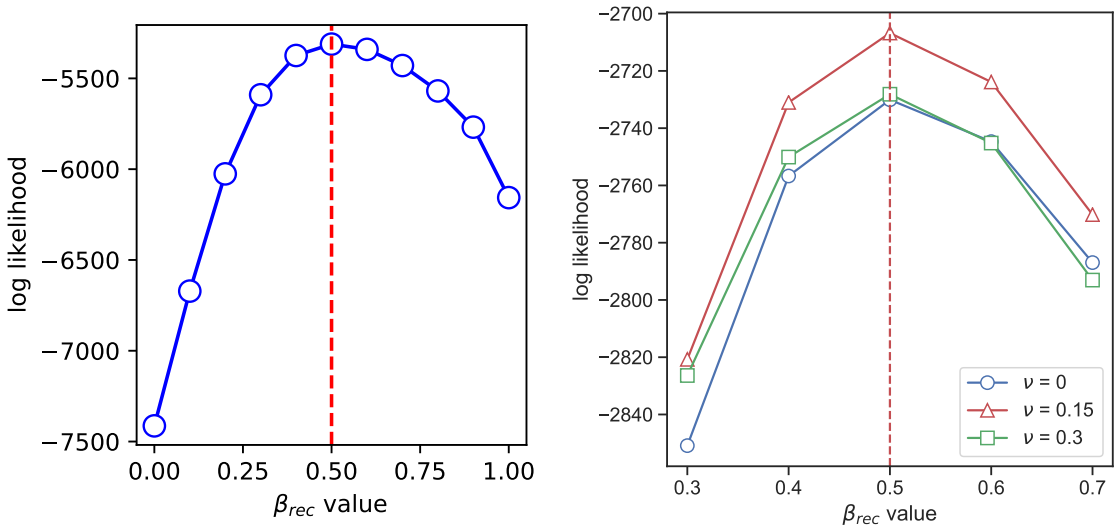

Figure 4: (a) Exercise 1. A visualization of model fitness for 11 model variants with different amounts of temporal reinstatement (β_{rec}). The x-axis shows β_{rec} for a given model, and the y-axis shows the log-likelihood for that model, after running a predictive simulation using the synthetic data described in the text. Larger values (closer to zero) indicate better model fitness. The red dashed line indicates the β_{rec} value (0.5) used to create the synthetic data. This coincides with the best-fitting model variant, indicating an ability to recover the generating parameter. (b) Exercise 2. Model fitness for model variants in which both temporal reinstatement (β_{rec}) and a neural scaling parameter (ν) are both manipulated. The synthetic recall data were generated by a model with $\beta_{rec} = 0.5$ and $\nu = 0.15$. The best-fitting model variant has these parameters, again indicating an ability to recover the generating parameters. See text for details.

to determine whether the parameters of a model are identifiable in a given simulation scenario. It is certainly possible for certain parameter settings to be recoverable, but for others to be ambiguous. For example, any parameter set that causes the model to produce no successful recalls will not be recoverable, as there are many parameter settings that can produce this particular failure state.

The code creates an array of β_{rec} values (called `B_rec_vals` in the code). As the code iterates through this array, it alters the `B_rec` parameter, runs a predictive simulation (with the `model.likelihood` function) and places the returned log-likelihood score into a results array (`logl`). Instead of performing a full search across all parameters, we only adjust the β_{rec} parameter, to determine whether the predictive model that matches the true generating value of β_{rec} (in this case 0.5) produces the best likelihood. The tutorial code will create a figure plotting the log-likelihood for each model variant. Figure 4a shows that indeed the data are best predicted by a model with β_{rec} set to 0.5.

Exercise 2: Fluctuating temporal reinstatement and synthetic neural data

In most published work using retrieved-context models, the temporal reinstatement parameter (β_{rec}) is set to a fixed value. It doesn't change from recall event to recall event. I don't think this is a strong claim of the model, in that I think it is reasonable to hypothesize that when a memory is retrieved, sometimes temporal reinstatement is more successful, and sometimes it is less successful. Usually, we don't have a principled way of knowing what those fluctuations are, so it makes sense to try to find a fixed value of β_{rec} that captures, in a sense, the expected amount of temporal reinstatement for a given recall event, as perhaps representative of the average recall event.

However, in the Kragel et al. (2015) study, we had some leverage to explore the possibility that temporal reinstatement indeed fluctuates from event to event. As described in the previous section, we examined the possibility that blood-flow to certain medial temporal lobe (MTL) brain structures reflects the degree of engagement of the temporal reinstatement mechanism. We tested this hypothesis by allowing the strength of a neural signal to control the recall-to-recall fluctuations in the β_{rec} parameter, and performing model comparison tests to see whether this improved the predictions made by the model.

Section 4 of the tutorial code allows you to examine a model in which the degree of temporal reinstatement fluctuates from recall event to recall event. Whereas the true underlying fluctuations in temporal reinstatement were unknowable in the Kragel et al. (2015) study (as they occurred in the participants' minds/cognitive systems), here we have created a simulated world in which we can perfectly know these fluctuations, because we construct them ourselves. This allows us to explore the conditions under which we can detect a correspondence between a noisy neural signal and a cognitive process. This approach certainly sidesteps many important complexities of the real world. For example, our predictive computational model is a rough and incomplete approximation of the true system generating the participant's behavior. But in this exercise the predictive computational model is a perfect match for the system generating the simulated participant's data. In any event, this approach allows us to demonstrate the logic and structure of this kind of analysis.

We first create synthetic neural signal values to represent signal recorded from the hippocampus of a participant. We create a matrix called `signal`, and fill it with fluctuations by drawing random numbers from a Normal distribution with mean = 0 and standard deviation = 1. We then create a field in the `recall` data structure called `hcmp` (for hippocampus), and fill it with these signal values. We create a `neural_scaling` parameter to be ν from Eq. 7, and set it to 0.15. In the next

section we try to recover this true neural scaling value during a parameter optimization process. We tell the code that the β_{rec} parameter is a dynamic parameter that is controlled by these neural signal values. Thus, for every recall event, the code will use the `hcmp` signal values to determine the value of `B_rec` (following Eq. 7). We now have a temporal reinstatement process that will vary randomly from recall event to recall event as we generate synthetic recall sequences. If you adjust the `neural_scaling` parameter, you can make these shifts in β_{rec} more subtle, or more dramatic, and see how this affects the results in the next section.

In Section 5, we use likelihood-based optimization to fit the model to the synthetic recall data generated in Section 4. In order to make the simulation more interesting, we embed the true `signal` used to control β_{rec} in random noise designed to obscure the signal. We introduce a `noise_weight` parameter controlling the relative contributions of the true signal and the random noise. This can be adjusted from to 0.0 for pure signal, to 1.0 for pure noise. The kind of noise one observes in functional MRI data doesn't likely follow a Normal distribution (Bullmore et al., 2001), but for the sake of the simplicity of the exercise we will stick with this. Following the procedure used by Kragel et al. (2015), we normalize the mixture of signal and noise at the trial level. For each trial, we apply a z-score transformation (subtracting off the mean of the observations and dividing by the standard deviation).

We then run likelihood-based optimization to attempt to recover the β_{rec} and neural scaling parameters used to generate our synthetic neural-behavioral data. The code performs a grid search, sweeping across 5 levels of β_{rec} (stepping from 0.3 to 0.7 in increments of 0.1; the generating value was 0.5), and 7 levels of the neural scaling parameter ν (stepping from 0.0 to 0.3 in steps of 0.05; the generating value was 0.15). Figure 4b shows the log-likelihood scores for several of these model variants, demonstrating that the model that is best able to predict the recall sequences is the one where both β_{rec} and ν match the original parameter values used to generate the synthetic recall sequences.

Finally, in Section 6 we run model comparison statistics to compare the predictive power of the different model variants with one another. Generally speaking, if you have a set of models, the one with the largest log-likelihood (i.e., closest to zero) makes the most accurate predictions. However, as theorists we also prefer models with fewer free parameters, for the sake of parsimony.

Consider two of the models used in the simulation exercises above, the neurally informed version of CMR where the ν parameter and the β_{rec} parameters are free to vary, and the neurally naive version where β_{rec} is free but ν is fixed at 0. The neurally naive model variant is nested within the

neurally informed model, as the neurally informed model contains the neurally naive model within its parameter space (i.e., when ν is set to 0). As such, it is not possible for the best-fitting neurally naive model to provide a larger log-likelihood (i.e., a better fit) than the best-fitting neurally informed model. Now, imagine that our observed neural signal was pure noise, i.e., its fluctuations don't correspond to the engagement of the temporal reinstatement mechanism. If there happen to be some spurious correspondences between the *pure noise* neural signal and the participant's behavior, this will lead to the neurally informed model yielding a better log-likelihood score than the neurally naive model. This potential advantage arises from the increased complexity of the neurally informed model relative to the naive model. Many model comparison methods provide a way to take this complexity into account, and apply a penalty to model fitness that scales with the number of free parameters.

One commonly used model comparison technique is the Akaike information criterion (AIC), which takes the number of free parameters and the number of data points into account (Wagenmakers & Farrell, 2004). This technique produces a score for each candidate model, which attempts to quantify the information loss when the probability distribution of the true generating model is approximated by the probability distribution associated with the candidate model (Burnham & Anderson, 2004). In this tutorial, we generated the data ourselves, so the probability distribution of the true model is actually knowable. However, in most applications, the data will be generated by actual participants with unknown true probability distributions.

Equation 8 shows the equation for a corrected form of AIC, called AIC_c. AIC_c includes an additive term generally penalizing more complex models ($2V$, where V indicates number of free parameters). A second additive term n indicates the number of data points.

$$AIC_c = -2logL + 2V + \frac{2V(V+1)}{(n-V-1)} \tag{8}$$

Here, we treat each recall event as a data point. As such, in our simulations above, the exact number of data points will depend on the stochastic recall processes implemented by the generative model. For a representative run of the generative model, the synthetic data contained 1246 events/data points.

Once an AIC score is calculated for each candidate model, the raw AIC scores can be transformed into Akaike weights. These weights can be interpreted as conditional probabilities representing the probability that each candidate model is the best model of the set. In this context, *best* is in terms

	n. param.	log(L)	AIC	wAIC
neurally naive model	1	-2730	5462	<0.00001
neurally informed model	2	-2707	5418	>0.99999

Table 1: Exercise 2. Log-likelihood scores and model comparison scores for representative runs of the neurally naive and neurally informed models. See text for details.

of the information theoretic definition of AIC mentioned above. Equations 9 and 10 show how these weights are calculated. First, a difference score is calculated for each model, where the best model's AIC score is subtracted from the candidate model's AIC score. Then these difference scores are used to calculate the relative support for each model.

$$\Delta_i(AIC) = AIC_i - minAIC \tag{9}$$

$$w_i AIC = \frac{exp(-\frac{1}{2}\Delta_i AIC)}{\sum_{k=1}^{K} exp(-\frac{1}{2}\Delta_k AIC)} \tag{10}$$

Given that the tutorial is built up around a stochastic generative model, if you run the tutorial code multiple times, each time you will get slightly different results. Table 1 provides results for a representative run of the tutorial simulations. The neurally informed model is usually preferred, which is to be expected, as we constructed the neural signal to have fluctuations corresponding to the fluctuations in temporal reinstatement.

We finish the tutorial with a demonstration of a potentially useful statistical tool, a permutation test (Hastie et al., 2001). The preceding demonstration used a neurally naive model as a baseline against which to compare the neurally informed model. One key difference between the neurally informed and neurally naive models is that in the neurally informed model, the β_{rec} parameter fluctuates from recall event to recall event, whereas in the neurally naive model, the parameter is stationary. One might hypothesize that the predictive advantage of the neurally informed model arises not because the variability in β_{rec} is specifically tracked by the fluctuations in the (synthetic) neural signal, but rather because it is generally advantageous to have this parameter vary as opposed to being stationary. We can call this the *generic variability* hypothesis. Our original neurally informed model represents a *specific variability* hypothesis in which the specific fluctuations observed on a particular trial are important. If the generic variability hypothesis is correct, we should be able to scramble the neural signal while preserving its general statistical characteristics, and preserve the predictive power of the model (i.e., get similar log-likelihood scores). If the specific

variability hypothesis is correct, scrambling the neural signal will harm the model's ability to predict behavioral performance.

Let's say we were also concerned that there could be temporal structure to our neural signal such that neighboring recall events tend to have similar neural signals associated with them. We know this isn't true in our synthetic data but it is a reasonable concern for neural recordings. In this case it wouldn't be fair to our assessment of the generic variability hypothesis to fully scramble the neural signals from recall event to recall event, as it would break this temporal structure. To address this concern, we partially scramble the synthetic neural signal at the level of trials. With 120 trials, we generate a permuted list of the integers from 1 to 120, and use these to rearrange the rows of the matrix carrying the synthetic neural signal. This preserves the structure of event-to-event fluctuations, while breaking the correspondence of these fluctuations to the events of a particular trial. A similar analysis was carried out by Kragel et al. (2015), where the goal was to determine whether a generic trend in the neural signal (e.g., on every trial the neural signal is gradually decreasing) could account for the neural-behavioral correspondence observed (it couldn't).

For this permutation test, we scramble the neural signal from trial to trial, and then re-calculate the goodness of fit (log-likelihood) of the model. We perform a number of iterations of this. For each iteration, we scramble trials, and re-calculate fit. Each iteration gives us a log-likelihood value, and together these form a distribution. We can then compare the original log-likelihood score (from Table 1: -2707) to this distribution. The proportion of scrambled scores that exceed the original score can be interpreted as a p-value. If the scrambling doesn't make a difference (as predicted by the generic variability model) then the original score should be somewhere in the middle of the permuted distribution. For the sake of efficiency, the permutation test in the tutorial code only runs 20 iterations. In our representative run, the original log-likelihood was greater than every value in the permutation distribution. This allows us to reject the generic variability hypothesis with $p < 0.05$. If you increase the number of iterations, you can get a more precise p-value.

Conclusion

In this chapter, we took a close look at the Context Maintenance and Retrieval (CMR) model of free recall. We examined different ways of using the model, including an approach in which the model (using a given parameter set θ) is used to calculate the likelihood of observing a given behavioral data set ($\mathbf{B}$, consisting of a set of recall sequences). This likelihood-based approach

allows one to optimize the model to find a set of parameters θ that maximize the model's ability to predict the set of recall events in $\mathbf{B}$. A given parameter set θ can also be used to generate synthetic recall sequences, allowing one to determine goodness of fit between observed summary statistics (calculated on B) and the same summary statistics calculated on the model-generated data. One of the powers of an event-level likelihood-based technique is that it allows a model to be sensitive to features of the data that might not be captured by standard summary statistics. This could include the trial-specific semantic identity of studied items (Morton & Polyn, 2016), the latency of individual responses (Osth & Farrell, 2019), or event-specific fluctuations in a neural signal (Kragel et al., 2015).

Kragel et al. (2015) used a likelihood-based modeling approach to examine the validity of different neural linking hypotheses, and to create model-based maps of the functional properties of neural signals in the medial temporal lobe. Turner et al. (2017) refer to this as a direct input approach, in which neural signal estimates are used to directly control the parameters of a cognitive model. This in turn affects the model's behavioral predictions. The tutorial simulations in this chapter provide an introduction to the retrieved-context model of free recall used by Kragel et al. (2015), and some of the techniques used to evaluate the model. Specifically, the tutorial demonstrates how one might demonstrate a functional correspondence between fluctuations in fMRI signal and a computation carried out by the retrieved-context model. Of course, it may be that the true computation carried out by these brain regions is substantially different from the temporal reinstatement mechanism implemented by the model. But the demonstration of a reliable functional correspondence between brain signal and cognitive mechanism suggests that the temporal reinstatement mechanism is worthy of further study.

Further exercises

- Try adjusting different model parameters and observe the effect on the serial position curve (SPC) and lag-CRP curve. For example, increasing parameter `P1` will increase the primacy effect of the SPC, decreasing `X2` will increase the overall number of items recalled, and altering `B_enc` will alter both the sharpness of the lag-CRP and the recency effect.

- The synthetic neural signal is embedded in noise, and the strength of the noise is controlled by the `noise_weight` parameter. Try increasing `noise_weight` to weaken the correspondence between the neural signal and model behavior. At some point, the permutation analysis will

no longer identify a statistically significant correspondence.

- The tutorial code contains variables controlling certain methodological characteristics of the simulated experiment: the number of participants, the number of trials per participant, and the number of items on a given study list. Try altering these variables to get a better sense of how they affect model performance. For example, you can increase study list length and see how this affects the primacy and recency effects seen in the SPC analysis figure.

References

Anderson, J. A., Silverstein, J. W., Ritz, S. A., & Jones, R. S. (1977). Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review*, *84*, 413–451.

Brown, G. D. A., Neath, I., & Chater, N. (2007). A temporal ratio model of memory. *Psychological Review*, *114*(3), 539–576.

Bullmore, E., Long, C., Suckling, J., Fadili, J., Calvert, G., Zelaya, F., ... Brammer, M. (2001). Colored noise and computational inference in neurophysiological (fMRI) time series analysis: Resampling methods in time and wavelet domains. *Human Brain Mapping*, *12*(2), 61–78.

Burnham, K. P., & Anderson, D. R. (2004, November). Multimodel inference understanding AIC and BIC in model selection. *Sociological Methods & Research*, *33*(2), 261–304. Retrieved 2013-07-25, from `http://smr.sagepub.com/content/33/2/261` doi: 10.1177/0049124104268644

Davelaar, E. J., Goshen-Gottstein, Y., Ashkenazi, A., Haarmann, H. J., & Usher, M. (2005). The demise of short-term memory revisited: Empirical and computational investigations of recency effects. *Psychological Review*, *112*, 3–42.

Dougherty, M., & Harbison, J. (2007). Motivated to Retrieve: How Often Are You Willing to Go Back to the Well When the Well Is Dry? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *33*(6), 1108.

Farrell, S. (2012). Temporal clustering and sequencing in working memory and episodic memory. *Psychological Review*, *119*(2), 223–271.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. New York: Springer-Verlag.

Healey, M. K., & Kahana, M. J. (2014). Is memory search governed by universal principles or idiosyncratic strategies? *Journal of Experimental Psychology—General*, *143*(2), 575–596.

Howard, M. W., Fotedar, M. S., Datey, A. V., & Hasselmo, M. E. (2005). The temporal context model in spatial navigation and relational learning: Toward a common explanation of medial temporal lobe function across domains. *PR*, *112*(1), 75–116.

Howard, M. W., & Kahana, M. J. (2002). A distributed representation of temporal context. *Journal of Mathematical Psychology*, *46*, 269–299.

Kahana, M. J. (2012). *Foundations of human memory* (1st ed.). New York, NY: Oxford University Press.

Kragel, J. E., Morton, N. W., & Polyn, S. M. (2015, February). Neural activity in the medial temporal lobe reveals the fidelity of mental time travel. *The Journal of Neuroscience*, *35*(7), 2914–2926.

Lehman, M. and Malmberg, K. J. (2013). A buffer model of memory encoding and temporal correlations in retrieval. *Psychological Review*, *120*(1), 155–189.

Lohnas, L. J., Polyn, S. M., & Kahana, M. J. (2015). Expanding the scope of memory search: Modeling intralist and interlist effects in free recall. *Psychological Review*, *122*(2), 337–363.

Luce, R. D. (1986). *Response times*. Oxford University Press.

Mallows, C. L. (1973). Sequential sampling of finite populations with and without replacement. *SIAM Journal on Applied Mathematics*, *24*(2), 164–168.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*(3), 419–57.

Miller, J. F., Weidemann, C. T., & Kahana, M. J. (2012). Recall termination in free recall. *Memory & Cognition*, *40*(4), 540–550.

Milner, B., Squire, L. R., & Kandel, E. R. (1998, March). Cognitive neuroscience and the study of memory. *Neuron*, *20*(3), 445–468.

Morton, N. W., & Polyn, S. M. (2016). A predictive framework for evaluating models of semantic organization in free recall. *Journal of Memory and Language*, *86*, 119–140.

Norman, K. A., & O'Reilly, R. C. (2003). Modeling hippocampal and neocortical contributions to recognition memory: A complementary learning systems approach. *Psychological Review*, *110*, 611–646.

Osth, A. F., & Farrell, S. (2019). Using response time distributions and race models to characterize primacy and recency effects in free recall initiation. *Psychological review*, *126*(4), 578.

Polyn, S. M., Norman, K. A., & Kahana, M. J. (2009). A context maintenance and retrieval model of organizational processes in free recall. *Psychological Review*, *116*(1), 129–156.

Purcell, B. A., Heitz, R. P., Cohen, J. Y., Schall, J. D., Logan, G. D., & Palmeri, T. J. (2010). Neurally constrained modeling of perceptual decision making. *Psychological Review*, *117*(4), 1113–1143.

Raaijmakers, J. G. W., & Shiffrin, R. M. (1981). Search of associative memory. *Psychological Review*, *88*, 93–134.

Schapiro, A. C., Turk-Browne, N. B., Botvinick, M. M., & Norman, K. A. (2017). Complementary learning systems within the hippocampus: a neural network modelling approach to reconciling episodic memory with statistical learning. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *372*, 20160049.

Sederberg, P. B., Howard, M. W., & Kahana, M. J. (2008). A context-based theory of recency and contiguity in free recall. *Psychological Review*, *115*(4), 893–912.

Tulving, E. (1993). What is episodic memory? *Current Directions in Psychological Science*, *2*(3), 67–70.

Turner, B. M., Forstmann, B. U., Love, B. C., Palmeri, T. J., & Van Maanen, L. (2017). Approaches to analysis in model-based cognitive neuroscience. *Journal of Mathematical Psychology*, *76*, 65–79.

Wagenmakers, E.-J., & Farrell, S. (2004, February). AIC model selection using akaike weights. *Psychonomic Bulletin & Review*, *11*(1), 192–196. Retrieved 2014-02-05, from `http://link.springer.com/article/10.3758/BF03206482` doi: 10.3758/BF03206482

Wilks, S. S. (1938, March). The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, *9*(1), 60–62. Retrieved 2014-12-04, from `http://www.jstor.org/stable/2957648`