The neural sampling hypothesis in dynamic environments

Jeremy Bernstein
Part III Project
Physics

Supervisor: Dr. Richard Turner
Co-supervisor: Dr. Máté Lengyel

May 18, 2016

Abstract

The brain must infer an understanding of the outside world from noisy sensory observations, like vision and hearing. A large body of evidence suggests the brain accomplishes this task via approximate Bayesian inference. Now the question is, how can Bayesian inference be performed neurally? In this project we investigate the *neural sampling hypothesis* that neural firing rates represent samples from the Bayesian posterior distribution.

The hypothesis is that a single neuron's firing rate obeys Markov chain dynamics. Classical Markov Chain Monte Carlo (MCMC) theory tells us that as the number of samples $n \to \infty$, a Markov chain converges to a unique stationary distribution, which we could set equal to the Bayesian posterior. But the brain needs to make decisions having only seen a finite number of samples. To go beyond classical MCMC theory, we explore different cost functions for targeting a posterior distribution based on finite n.

We find that a cost function measuring the Kullback-Leibler divergence of the chain's joint distribution from n copies of the target distribution pressures the chain to decouple rather than hit the correct target. Another cost function involves taking the mean squared error (MSE) of Monte Carlo averages. We find that this cost function naturally encodes the tradeoff between decoupling and targeting the right distribution for finite n. We also find that as $n \to \infty$ this cost function recovers the classical goal of Bayesian inference via MCMC: hitting the right stationary distribution. We also link an MSE cost function in the limit that $n \to \infty$ to the KL divergence in the limit of small cumulant discrepancy, providing a more information-theoretic justification for only minimising the MSE of the first few moments.

Finally we propose two testable predictions of the neural sampling hypothesis based on our work: first, under certain conditions, decisions based on a sequence of posterior samples should place more trust in the first and last samples, implying a time-lagged correlation between sampler firing rate and decision outcome. Second, variance in neural firing rates is problematic when making quick decisions (based on finite n), but irrelevant when making slower decisions ($n \to \infty$). Therefore under the sampling hypothesis we'd expect to see less variance in firing rate during more rapid decision-making.

Contents

Notation

NSH	Neural sampling hypothesis	
PPC	Probabilistic population codes	
MCMC	Markov Chain Monte Carlo	
KL$(q\|\|p)$	KL divergence between distributions q and p	$\int dx\, q \log \frac{q}{p}$
$\hat{\lambda}$, $\tilde{\lambda}$	Estimators of quantity λ	
MSE	Mean squared error of estimator	$\left\langle (\tilde{\lambda} - \lambda)^2 \right\rangle$
$x \sim \mathcal{N}(\mu, \sigma^2)$	x distributed normally with mean μ and variance σ^2	
iid	independent and identically distributed	

Table 1: Important notation used in the text.

1 Introduction and motivation

1.1 The brain as a computational engine

It is an exciting time for neuroscience. The recent advent of multielectrode arrays allows massive amounts of neural data to be collected in parallel in vivo [Spira and Hai, 2013]. Hypotheses about brain function can be rigorously tested, and the tools of statistical physics are just right to uncover some of the basic organising principles that govern how our brains compute.

In the brain billions of *neurons* are connected by trillions of *synapses*. One neuron firing influences the probability that other synaptically connected neurons also fire. This is the mechanism by which the brain computes.

Various models of neural computation exist at different levels of abstraction [Dayan and Abbott, 2005, ch. 7]. From here on in, we will assume that neurons are well-modelled by each having a continuous *firing rate*. This coarse-graining throws away information contained in the delay between individual spikes. We will further assume there is a characteristic time for the firing rate of a given neuron to change, so we will trade the continuous rate $x(t)$ for a discretised time series of rates $\{x_1, x_2, ..., x_t\}$. We rescale the discretised rates to lie in the range $-\infty$ to ∞.

1.2 Bayesian inference in the brain

The brain's current belief about the world arises from continual integration of new sensory information into its own prior belief. The prior belief is gained from all past observations. In this project, we ask how such belief about the world might be maintained.

To give a concrete example, imagine programming a robot to keep track of its position, x, whilst it moves around its environment. Every time the robot makes a movement δx it should increment its position variable to $x + \delta x$, constituting integration of new information with prior belief. But this neglects an important factor: the robot can't be certain it moved *exactly* δx. In fact, it only has the noisy measurement $y \approx \delta x$. The robot should settle for keeping track of its position and associated uncertainty by storing a probability distribution $p(x)$. The appropriate

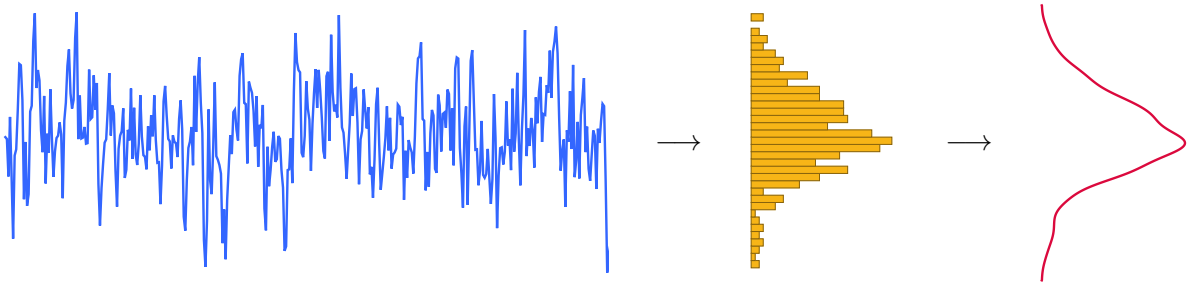


Figure 1: How a firing rate encodes a distribution according to the neural sampling hypothesis. The firing rate represents samples from a distribution. Binning the samples, we can see the distribution emerging. The righthand panel was obtained by smoothing the histogram. In fact, this data was generated from a linear Gaussian Markov chain: for longer runs of the chain, the distribution would look more and more Gaussian.

way to update $p(x)$ in the face of new information y is by using Bayes' rule:

$$\mathrm{p}(x|y) = \frac{\mathrm{p}(y|x)\mathrm{p}(x)}{\mathrm{p}(y)} = \frac{\mathrm{p}(y|x)\mathrm{p}(x)}{\sum_{x'} \mathrm{p}(y|x')\mathrm{p}(x')} \tag{1.1}$$

Provided the robot can model observation noise $p(y|x)$ then at least in principle it can apply Bayes' rule to track its position with associated uncertainty.

Much evidence suggests that the brain uses Bayesian inference, just as we programmed the robot to. Now the important question is *how is Bayesian inference implemented neurally?* A neural Bayesian inference algorithm must overcome certain hurdles:

1. how can probability distributions be represented neurally in the first place? Neurons have inherent noise, which could be a help or a hindrance.
2. how can such a neural probability distribution be updated in the face of new observations?
3. exact Bayesian inference is computationally costly – how does the brain do it so fast?

Part of the difficulty can be overcome by noting that we'd expect approximate Bayesian inference to be faster than exact Bayesian inference. The brain surely settles for the speed-accuracy tradeoff best for survival. The neural sampling hypothesis is one candidate theory for how the brain could perform approximate inference.

1.3 The neural sampling hypothesis

The hypothesis says that neural firing rates *sample* from the approximate Bayesian posterior distribution.

Let's explain what this means by example. Under the hypothesis, 3 neurons are needed to encode a distribution over 3D position, call it $q(x, y, z)$. At any time, the 3 firing rates are a specific position $(x,\ y,\ z)$ which is drawn from the distribution $q(x, y, z)$. The distribution should approximate the exact Bayesian posterior distribution, which is written $p(x, y, z|y_{1:t})$ and is conditioned on all prior observations $y_{1:t}$. If all three firing rates linger around zero with small variance, this constitutes a strong belief that the position is close to the origin. This idea is illustrated in Figure 1.

The idea is elegant for several reasons:

1. only d neurons are needed to represent arbitrarily complicated d-dimensional distributions – although of course more neurons are needed to update these d neurons;
2. things fit naturally within the framework of neural network computation, and intrinsically use noise rather than regarding noise as a hindrance;
3. the hypothesis draws on the well established theory of Markov Chain Monte Carlo (MCMC) as a means to dynamically update the neurons, to draw samples from the posterior.

The fundamental drawback of neural sampling is that it is limited by time, since many samples need to be generated to well-describe the full distribution.

1.4 Our investigation

Typical existing work in performing Bayesian inference via MCMC involves setting up a Markov chain such that the n^{th} sample draws from the true posterior distribution in the limit that $n \to \infty$. Other work on the neural sampling hypothesis treats the case where *only one* sample is drawn per new observation of the world, corresponding to the case $n = 1$.

By comparing timescales of the natural world to neural timescales, we argue that n of order 10 is appropriate for neural sampling algorithms. We also argue that the brain cares about the distribution of all n samples, rather than just the marginal distribution of the last one. Therefore in this project we seek a normative means of setting Markov chain parameters for the joint distribution of n samples that is most useful for the brain.

In Section 2 we give a statistical primer and review the existing literature on Bayesian inference in the brain to motivate our own line of investigation. In Section 3 we fully describe the problem we set out to solve, and give our theoretical results. In Section 4 we discuss our results and suggest directions for further work.

Appendices A, B and C derive results used in the text. Appendix D provides additional context on the link between statistical physics and neuroscience.

2 Literature review and background theory

2.1 Statistical primer: Markov chains and cost functions

A Markov chain is a sequence, or *chain*, of random variables $\{x_i\}$ such that x_{i+1} only depends on x_i: the future depends on the present. The fundamental result of Markov Chain Monte Carlo (MCMC) is that the distribution of x_∞ is unique and independent of the initial condition of the chain [MacKay, 2002]. We say that samples drawn from the chain converge to a *stationary* or *equilbrium* distribution.

One Markov chain often used in this report is the linear Gaussian chain:

$$x_{i+1} = ax_i + \sigma\epsilon_i \tag{2.1}$$

Here $0 < a < 1$ and $\epsilon_i \sim \mathcal{N}(0, 1)$. The next value of the chain is just a noisy, scaled copy of the previous one. In Appendix A.1 we show that for this chain the stationary distribution is Gaussian:

$$x_\infty \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-a^2}\right) \tag{2.2}$$

Typically we are free to vary the Markov chain's parameters, ψ, to affect some related distribution q_ψ (such as the stationary distribution). In this report we are concerned with trying to match $q_\psi(x)$ to some target distribution $p(x)$. We can do this by minimising with respect to ψ the Kullback-Leibler (KL) divergence

$$KL(q_\psi||p) = \int_{-\infty}^{\infty} dx\, q_\psi \log \frac{q_\psi}{p} \tag{2.3}$$

The KL divergence has two important properties making it suitable for matching distributions:

1. $KL(q_\psi||p) = 0$ if and only if $q_\psi(x)$ exactly matches $p(x)$
2. the larger $KL(q_\psi||p)$ is, the worse the distributions match

If we are able to draw samples $(x_1, ..., x_n)$ from $q_\psi(x)$ then we can match it to the target $p(x)$ in another way. We note that the average

$$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} f(x_i) \tag{2.4}$$

is a Monte Carlo estimator of the quantity $\lambda = \int dx\, p(x)f(x)$. Seeing as the samples were drawn from q_ψ and not p, the estimator will be best in the case that q_ψ exactly matches p. The error of the estimator is measured via the mean squared error:

$$MSE(\hat{\lambda}) = \left\langle (\hat{\lambda} - \lambda)^2 \right\rangle \tag{2.5}$$

Defining the bias $b = \left\langle \hat{\lambda} \right\rangle - \lambda$, and variance $V = \left\langle \hat{\lambda}^2 \right\rangle - \left\langle \hat{\lambda} \right\rangle^2$, we can write

$$MSE(\hat{\lambda}) = b^2 + V \tag{2.6}$$

as derived in Appendix A.3. Therefore minimising both the bias and variance of $\hat{\lambda}$ with respect to Markov parameters ψ will pressure $q_\psi(x)$ to approximate $p(x)$.

Quantity	Fluctuates on timescale
Feature in natural world	500ms
Neuron's firing rate	50ms

Table 2: Very approximate timescales important for neural inference of outside world events. [Murray et al., 2014]. An individual neuron has the chance to explore a handful of independent firing rates before the outside world noticably changes. In the neural sampling framework, this allows a sampling algorithm to draw a finite number n of samples per time step of the outside world. This data suggests an n of order 10.

2.2 Evidence for the neural sampling hypothesis

A large body of evidence suggests the brain uses optimal statistical inference, e.g. [Ernst, 2002, Kording and Wolpert, 2004]. There are two rival explanations for the neural implementation of Bayesian inference:

1. PPC – *probabilistic population codes* [Ma et al., 2006] – parameters of the distribution are encoded across many neurons. This is temporally efficient since the full distribution is encoded at all times.
2. NSH – *neural sampling hypothesis* [Fiser et al., 2010] – a single neuron samples from the distribution. This uses less space than PPC, but is time limited since many samples are needed to represent the full distribution.

Much psychophysical evidence supports NSH [Gershman et al., 2012, Lieder et al., 2012, Vul et al., 2014]. Some important research has gone into making neural predictions of NSH, such as [Berkes et al., 2011, Hennequin et al., 2014, Haefner et al., 2016].

2.3 Timescales of existing neural sampling algorithms

NSH stipulates that the firing rates of a neuron represent samples from a distribution. But how can the firing rate target a particular distribution? A nice idea is that the firing rates follow Markov dynamics, and therefore target the stationary distribution of the Markov chain. Much existing work is then concerned with trying to match the posterior distribution to the stationary distribution of the chain, or sometimes to the distribution of the n^{th} sample [Salimans et al., 2015], with the idea that results will become exact as $n \to \infty$.

Another common paradigm for Bayesian inference under NSH is described in Figure 2. The brain is trying to recognise the state of the world x_t from a time series of noisy sensory observations $\{y_1, ..., y_t\}$ [Greaves-Tunnell, 2015, Kutschireiter et al., 2015]. To apply Bayes' rule to perform inference, the brain must have learnt an internal model for $p(y_t|x_t)$, known as the generative model. But it is always assumed that the sampling algorithm works at the same 'clock rate' as the generative model. Drawing one sample per step of the generative model can be thought of as MCMC in the limit that $n = 1$.

So previous work operates on two time scales: $n = 1$ and $n \to \infty$. In Table 2 we argue that the biologically relevant timescale is in fact finite n of order 10 – the brain works faster than the outside world. Our work differs to [Salimans et al., 2015] as we consider the distribution of all n samples as being important, rather than just the distribution of the last one.

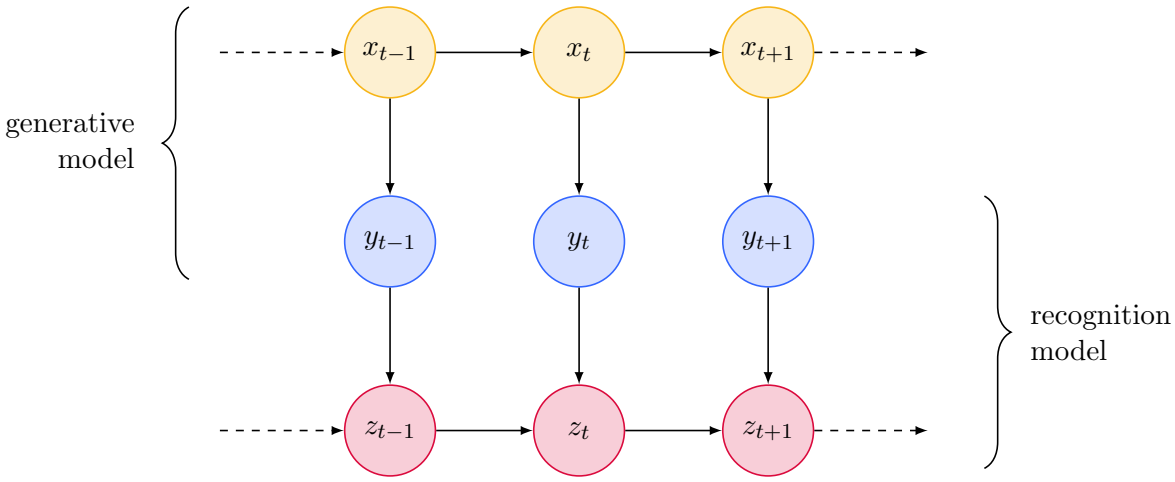


Figure 2: The model considered in [Greaves-Tunnell, 2015]. x_t is the unknown state of the world at time t, which evolves according to some noisy dynamical law. An observation y_t, such as a 2D retinal image of the 3D world, is generated noisily from x_t. The dynamical evolution of x_t and the generation of y_t is known as the *generative model*, as together they form the brain's understanding of how sensory information is generated. The *recognition model* consists of using observation y_t and previous guess z_{t-1} to form a new guess z_t of the unknown x_t. Ideally z_t is a sample from the true posterior distribution over x_t given all past observations y_1 to y_t.[1]

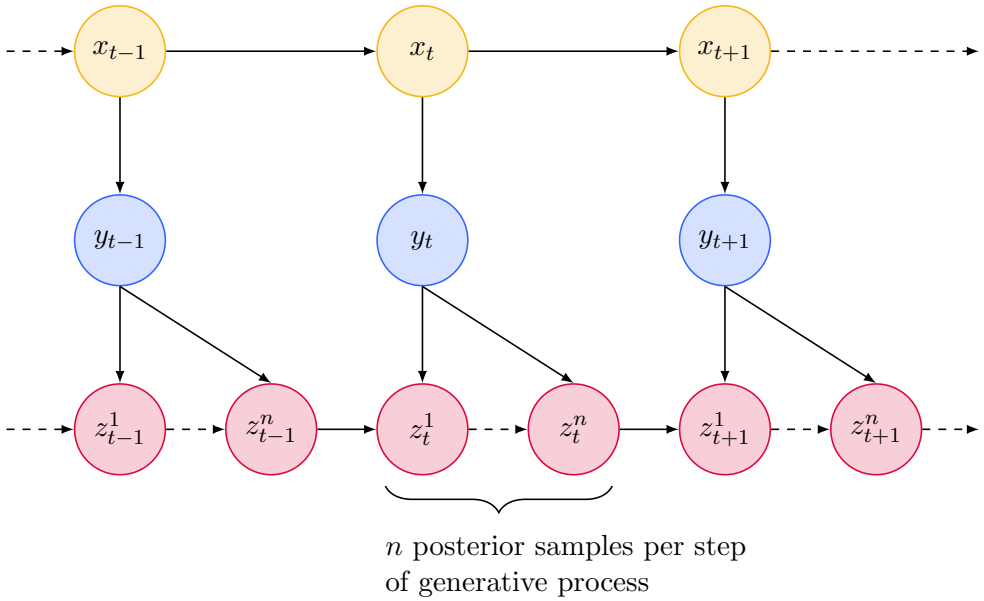


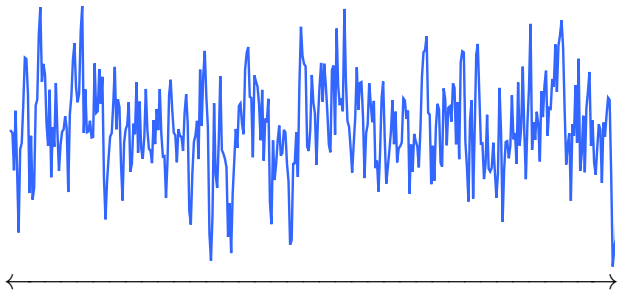
Figure 3: Our model. x_t represents the unknown state of the world at time t. y_t is an observation generated from x_t, such as a 2D retinal image of the 3D world. z_t is a guess at the state x_t having seen y_t. In contrast to [Greaves-Tunnell, 2015], we assume n posterior samples can be produced per observation of the outside world, corresponding to the notion of 'overclocking' the recognition model. We investigate how to set up a Markov chain to generate these n samples.[1]

[1]Note that in these figures, the arrows leading into z_t signify what is used to directly compute z_t in the model, and *not* the statistical dependence. The distribution of z_t should statistically depend on *all* observations y_1 to y_t.

3 Theoretical contributions

3.1 The relevant neural timescale

In Section 2 we noted that previous work on MCMC for Bayesian inference focuses on drawing n samples from a Markov chain, where either $n \to \infty$ or $n = 1$. Clearly $n \to \infty$ cannot be achieved by the brain, since it must make decisions in finite time. But equally $n = 1$ is too small since neural firing rates can change faster than the typical timescale of events in the outside world. In Table 2 we argue that a neuron can explore n of order 10 firing rates before the outside world noticeably changes.



a neural trace of $n \approx 100$ samples

Given that a neural sampler can use n samples to target the Bayesian posterior distribution, the question is how should those samples be generated? We want a cost function to measure the closeness between n Markov samples and the posterior distribution. A key difference of our work to prior work is that we argue that *all* n samples are useful to the brain, rather than just the marginal distribution over the last sample.

3.2 Finding the right cost function for n samples

We imagine drawing a finite number of samples n from a Markov chain. The full distribution over the n samples $\mathbf{x} = (x_1, x_2, ..., x_n)$ can be written

$$Q(\mathbf{x}) = q(x_1) \prod_{i=2}^{n} q(x_i|x_{i-1}) \tag{3.1}$$

which is directly encoding the Markov property. We want our n samples to lie 'close' to a target distribution $p(x)$. We need to clarify what notion of 'closeness' we should use.

Suppose we are trying to target $p(x)$, which is distributed $\mathcal{N}(0, \tau^2)$, by drawing a finite number of samples n from the linear Gaussian chain – Equation 2.1: $x_{i+1} = ax_i + \sigma\epsilon_i$. We assume that σ models background neural noise and can't be varied. The task is to vary a to make the samples target $p(x)$.

The stationary distribution of this Markov chain is $\mathcal{N}(0, \frac{\sigma^2}{1-a^2})$. Therefore choosing a to satisfy $\frac{\sigma^2}{1-a^2} = \tau^2$ should get the last sample close to the target distribution. On the other hand this might make consecutive samples highly dependent. Letting $a \to 0$ (decoupling the chain) would solve this problem by making consecutive samples completely independent.

We need to find a cost function that naturally encodes this tradeoff between:

1. making consecutive samples independent;
2. making the marginal distribution of a given sample close to $p(x)$.

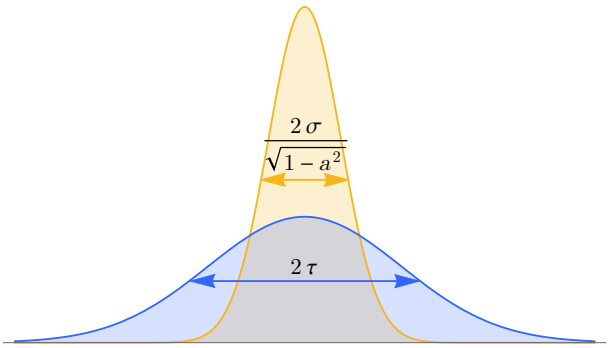


Figure 4: Comparing a target Gaussian with variance τ^2 to the stationary distribution of a linear Gaussian Markov chain ($x_{i+1} = ax_i + \sigma\epsilon_i$) which is Gaussian with variance $\frac{\sigma^2}{1-a^2}$. Whilst increasing a towards 1 would push the stationary distribution closer to the target distribution as desired, it would also have the negative effect of making consecutive samples more dependent. The choice of a should be decided by some tradeoff of these competing factors.

3.3 KL divergence cost function

The ideal behaviour of the sampler would be to draw n iid samples from the exact target distribution. That would correspond to a single n dimensional sample from

$$P(\mathbf{x}) = \prod_{i=1}^{n} p(x_i) \tag{3.2}$$

Motivated by this, we considered the cost function

$$\boxed{C_1 = \mathrm{KL}\left[Q(\mathbf{x})||P(\mathbf{x})\right]} \tag{3.3}$$

in the hopes that minimising this KL divergence with respect to the Markov chain parameters should naturally control the tradeoff between sample independence and sample accuracy.

We discovered that this is not in general the case. In fact for the linear Gaussian chain this cost function completely favours decoupling. We can see this by using the following result (proved in Appendix A.2):

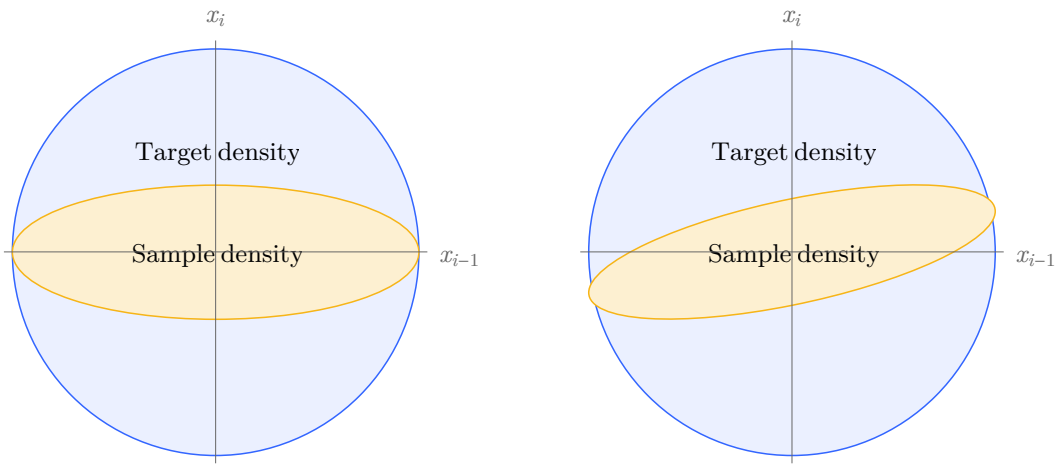
$$\mathrm{KL}\left[Q(\mathbf{x})||P(\mathbf{x})\right] = \mathrm{KL}\left[q(x_1)||p(x_1)\right] + \sum_{i=2}^{n} \left\langle \mathrm{KL}\left[q(x_i|x_{i-1})||p(x_i)\right]\right\rangle_{q(x_{i-1})} \tag{3.4}$$

This just says that the KL divergence between the full chain and n iid samples from the target distribution can be broken up into two parts: the first term measures the distance between the starting value of the chain and the target. The terms in the summation measure the distance between the Markov transition and the target, averaged over the marginal distribution of the previous sample.

Let's try to minimise Equation 3.4 with respect to a for the chain $x_{i+1} = ax_i + \sigma\epsilon_i$. If we assume there is no freedom in choosing the chain's initial value, then the important terms are of the form $\left\langle \mathrm{KL}\left[q(x_i|x_{i-1})||p(x_i)\right]\right\rangle_{q(x_{i-1})}$. For this chain, $q(x_i|x_{i-1})$ is $\mathcal{N}(ax_{i-1}, \sigma^2)$. But clearly since the target distribution is Gaussian with zero mean, $q(x_i|x_{i-1})$ had also better be Gaussian with zero mean. Therefore every term under the average is minimised by setting $a = 0$, and this cost function favours decoupling. The argument is illustrated in Figure 5.

3.4 Mean squared error of a Monte Carlo average

We showed in the previous section that a KL cost function that pressures the chain's samples to be drawn iid from the target distribution does not in general yield the desired tradeoff between



Better KL overlap with $a = 0$ Worse KL overlap with $a = 0.2$

Figure 5: Comparing standard deviation ellipses for target distribution (blue) and Markov chain samples (orange). The target is $\mathcal{N}(0, 1)$ and the chain is $x_{i+1} = ax_i + \epsilon_i/3$ where $\epsilon_i \sim \mathcal{N}(0, 1)$. The joint target density is $P = p(x_i)p(x_{i-1})$. The joint Markov chain density is $Q = q(x_i|x_{i-1})q(x_{i-1})$. We assume the $(i-1)^{th}$ marginal $q(x_{i-1})$ is close to the target density. With $a = 0$ (left) the KL divergence is better (smaller) than with $a = 0.2$ (right). So the cost function $\mathrm{KL}(Q||P)$ pressures the chain to decouple, and the stationary distribution will *not* be close to the target.

sample independence and sample accuracy. In this section we will propose a new cost function, by thinking about how the brain might actually want to *use* the samples.

The idea is motivated by the schematic given in Figure 6. We imagine that another brain region sees n samples and needs to make a decision (such as planning a motor action). All that matters is that taken together the n samples well-characterise the feature of the distribution on which the decision is based. It does not directly matter whether consecutive samples are independent.

One way to encode this idea is by adopting the view that the decision network just wants the samples to compute some Monte Carlo average:

$$\hat{\lambda}_\phi = \frac{1}{n}\sum_{i=1}^{n} \phi(x_i) \tag{3.5}$$

Here the sum is over the n samples, and $\phi(x)$ encodes the feature of the distribution in which the decision network is interested (perhaps the mean or the second moment). The form of this expression makes it obvious that the order of the samples or the degree of independence of consecutive samples is not important. All that matters is how closely $\hat{\lambda}$ approximates the integral $\lambda_\phi = \int dx\, p(x)\phi(x)$.

This naturally prompts a mean squared error (MSE) cost function of the form

$$\boxed{C = \left\langle (\hat{\lambda}_\phi - \lambda_\phi)^2 \right\rangle} \tag{3.6}$$

and begs the question: how should we choose $\phi(x)$? We discuss this question in the next section.

3.5 Choosing the right function to Monte Carlo average

We are considering cost function $\left\langle (\hat{\lambda}_\phi - \lambda_\phi)^2 \right\rangle$: the mean squared error of a Monte Carlo average based on n samples from a Markov chain.

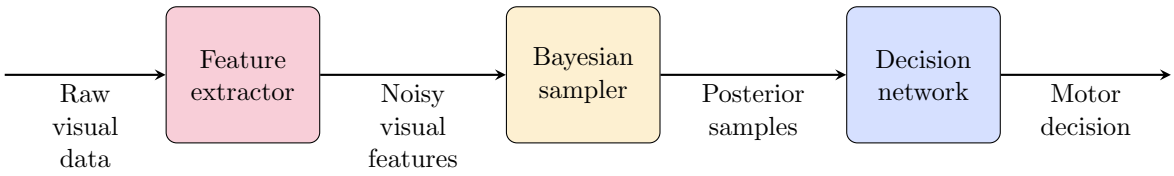


Figure 6: A schematic neural algorithm for making a motor decision based on noisy visual input. The feature extractor is a feedforward neural network which extracts features (such as objects and positions) from the raw pixel data. The Bayesian sampler is a recurrent neural network which combines the noisy visual features with prior belief. The decision network watches the output samples and uses them to plan motor action.

The question is: which function $\phi(x)$ should we Monte Carlo average? We can rephrase this question as: what property of the target distribution does the brain want to be able to estimate most accurately? When making decisions in the face of uncertainty, we argue that it is most important to know the average belief and the associated variance – so the lowest moments should be most important: $\phi(x) = x$ and $\phi(x) = x^2$.

Defining $\hat{\lambda}_k$ to be the Monte Carlo estimate of the k^{th} target moment

$$\hat{\lambda}_k = \sum_{i=1}^{n} \frac{x_i^k}{n} \sim \int dx\, p(x) x^k \tag{3.7}$$

we consider the cost function

$$C = \sum_{k=1}^{\infty} f(k) MSE(\hat{\lambda}_k) \tag{3.8}$$

where $f(k)$ is some decaying function, to encode the fact that it is most important to estimate the lowest moments well.

Using the bias-variance decomposition for MSE, we can rewrite this cost function as

$$C = \sum_{k=1}^{\infty} f(k) b_k^2 + \sum_{k=1}^{\infty} f(k) V_k \tag{3.9}$$

where b_k is the bias and V_k is the variance of $\hat{\lambda}_k$.

The first term encodes the bias of all the estimators. This term will be minimised by making the marginal distribution of each sample lie close to the target distribution. The second term encodes the variance of the estimators. This term will be minimised by decoupling the chain, since then the chain will converge fastest to the stationary distribution. This can be seen intuitively: if the Markov chain is strongly coupled then it will take a long time to traverse the state space, and using a short run of the chain to make estimates will have a high variance.

Therefore this cost function *exactly* encodes our desired tradeoff between hitting the right target distribution (the bias term) and decoupling the chain (the variance term). We demonstrate this for the linear Gaussian chain in Appendix B.

Also consider the limit that $n \to \infty$. In this limit $\hat{\lambda}_k$ tends exactly to the k^{th} moment of the chain's stationary distribution[2]. Therefore the variance $V_k \to 0$, and the bias $b_k \to \epsilon_k$ where we define ϵ_k to be the k^{th} *moment discrepancy* – i.e. the difference between the k^{th} moments of the chain's stationary distribution and the target distribution.

So the bias-variance tradeoff disappears in this limit, and the cost function reduces to

$$\lim_{n \to \infty} C = \sum_{k=1}^{\infty} f(k) \epsilon_k^2$$

[2]This relies on the key result that for long enough runs of a Markov chain, time averages are equivalent to ensemble averages.

Minimising this cost function amounts to setting the chain's stationary distribution equal to the target distribution, and we have recovered the classical goal of Bayesian inference via MCMC.

In Appendix C we derive a very similar expression involving the KL divergence. Assuming $q(x)$ is $\mathcal{N}(\mu, \sigma^2)$, and $p(x)$ only differs from $q(x)$ weakly, then

$$KL(q||p) = \frac{1}{2}\sum_{k=1}^{\infty} \frac{\eta_k^2}{k!\sigma^{2k}} \tag{3.10}$$

Here the η_k is the k^{th} *cumulant* discrepancy, in contrast to ϵ_k the k^{th} moment discrepancy. Cumulants are simple polynomial functions of the moments: for example the first cumulant is the mean, the second is the variance, the third is the skewness, etc. The results of this analysis in full generality, using the theory of Edgeworth series, are given in Appendix C.

Since the KL divergence is an intrinsically information theoretic quantity, we can use the general functional form of Equation 3.10 to motivate a choice of decay function $f(k) = \frac{1}{k!}$.

Finally we can give our ultimate cost function, and summarise its properties.

$$\boxed{C_2 = \sum_{k=1}^{\infty} \frac{1}{k!} MSE(\hat{\lambda}_k)} \tag{3.11}$$

1. for finite n, the cost function encodes the tradeoff between hitting the right target distribution, and successive samples being independent;
2. as $n \to \infty$, the cost function only cares about matching the stationary distribution to the target;
3. the cost function cares less about estimating higher moments.

3.6 Freedom in decoding weights

In this section we wish to study what freedom exists in a brain region that is *using* the Markov samples, perhaps to make decisions as in Figure 6. To formalise this problem, we ask what are the optimum *decoding weights* w_i in the Monte Carlo average

$$\tilde{\lambda}_\phi = \sum_{i=1}^{n} w_i \phi(x_i) \tag{3.12}$$

when estimating $\lambda = \int dx\, p(x)\phi(x)$?

To simplify the problem we assume that n samples are drawn from the chain *starting at equilibrium*, and that the stationary distribution of the chain *is the target distribution* $p(x)$. This implies our analysis will only probe the effect of dependence between samples. It also leads to mathematical simplicity, since then the condition for $\tilde{\lambda}_\phi$ being an unbiased estimator of λ_ϕ is just $\sum_i w_i = 1$. The assumption is biologically reasonable in the case when no new information has been observed by the brain for long enough for the chain to reach stationarity.

We wish to minimise the mean squared error (MSE) of $\tilde{\lambda}_\phi$ subject to the constraint that it is an unbiased estimator of λ_ϕ. First we decompose the mean squared error into the bias, $b = \langle\tilde{\lambda}_\phi\rangle - \lambda_\phi$, and variance, $V = \langle(\tilde{\lambda}_\phi - \langle\tilde{\lambda}_\phi\rangle)^2\rangle$ (as in Appendix A.3)

$$\text{MSE} = \left\langle(\tilde{\lambda}_\phi - \lambda_\phi)^2\right\rangle = V + b^2 \tag{3.13}$$

Therefore minimising the MSE subject to the constraint of unbiasedness ($b = 0$) is equivalent to minimising the variance subject to the same constraint. We introduce the Lagrange multiplier

Algorithm 1 Finding optimal decoding weights. We seek the weights w_i which minimise the variance in the Monte Carlo average $\tilde{\lambda}_\phi = \sum_{i=1}^n w_i \phi(x_i)$ as an estimator of $\lambda = \int dx\, p(x)\phi(x)$, subject to the constraint of zero bias. We assume the samples in the Monte Carlo average are drawn from a Markov chain that begins at equilibrium, and that the equilibrium distribution of the chain is the target distribution $p(x)$.

1: Compute autocovariance matrix of the samples: $C_{ij} = \langle \phi(x_i)\phi(x_j)\rangle - \langle \phi(x_i)\rangle\langle \phi(x_j)\rangle$
2: Invert the autocovariance matrix to find C^{-1}
3: Multiply the vector of 1's by C^{-1} to find $\boldsymbol{w}^* = C^{-1}(1, 1, 1, ...1)$ (unnormalised)
4: Normalise with the 1-norm to find decoding weights $\boldsymbol{w} = \boldsymbol{w}^*/|\boldsymbol{w}^*|_1$

μ and minimise with respect to w_i and μ

$$\mathcal{L} = \left\langle (\tilde{\lambda}_\phi - \langle \tilde{\lambda}_\phi \rangle)^2 \right\rangle - \mu \left(\sum_i w_i - 1 \right) \tag{3.14}$$

Substituting in the expression for $\tilde{\lambda}_\phi$ this reduces to

$$\mathcal{L} = \sum_{ij} w_i w_j \left[\langle \phi(x_i)\phi(x_j)\rangle - \langle \phi(x_i)\rangle\langle \phi(x_j)\rangle \right] - \mu(\sum_i w_i - 1) \tag{3.15}$$

If we define the autocovariance matrix $C_{ij} = \langle \phi(x_i)\phi(x_j)\rangle - \langle \phi(x_i)\rangle\langle \phi(x_j)\rangle$, extremising this Lagrangian yields a simple expression for the decoding weights: $\boldsymbol{w} \propto C^{-1}(1, 1, 1, ..., 1)$. We summarise the simple algorithm for finding the decoding weights in Algorithm 1.

For the special case of computing the mean ($\phi(x) = x$) of the output of the linear Gaussian chain ($x_{i+1} = a x_i + \sigma \epsilon_i$), the autocovariance (a Toeplitz matrix) and its inverse are

$$C = \frac{\sigma^2}{1-a^2} \begin{pmatrix} 1 & a & a^2 & \cdots & a^{n-1} \\ a & 1 & a & & \vdots \\ a^2 & a & 1 & & a^2 \\ \vdots & & & \ddots & a \\ a^{n-1} & \cdots & a^2 & a & 1 \end{pmatrix} \qquad C^{-1} = \frac{1}{\sigma^2} \begin{pmatrix} 1 & -a & 0 & \cdots & 0 \\ -a & 1+a^2 & -a & & \vdots \\ 0 & -a & 1+a^2 & & 0 \\ \vdots & & & \ddots & -a \\ 0 & \cdots & 0 & -a & 1 \end{pmatrix} \tag{3.16}$$

Following Algorithm 1 yields weights $\boldsymbol{w} \propto (1, 1-a, 1-a, ..., 1-a, 1)$ – illustrated in Figure 7. This is a nice result: it says to reduce the error in the mean of a Markov chain run from equilibrium, trust more the first and last samples than the ones in between. The intuition behind this result is that the first and last samples are the most decoupled and therefore contain the most useful information.

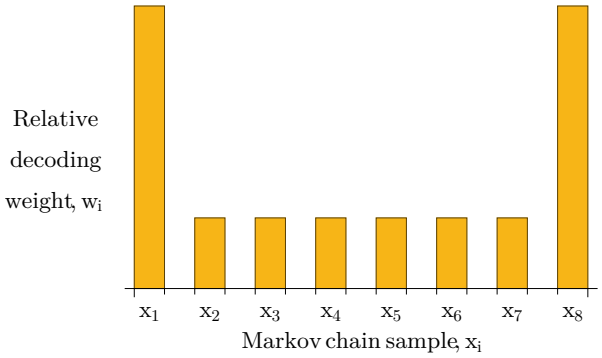


Figure 7: The optimal decoding weights for $n = 8$ samples from the linear Gaussian chain with coupling strength $a = 0.75$. The vertical size of the bars represents the relative size of the weights. The weights are used in computing the Monte Carlo average $\tilde{\lambda} = \sum_{i=1}^{8} w_i x_i$. More weight is put on the first and last samples from the chain than from the samples in the middle. Intuitively this is because the first and last samples are the most statistically independent, whereas the ones in the middle are more correlated.

4 Discussion and conclusion

In this project we investigated how Bayesian inference might be performed neurally. The neural sampling hypothesis is one suggested answer. It says that neural firing rates follow MCMC dynamics to sample from the Bayesian posterior distribution.

Prior work in this area assumed that the neural sampling algorithm runs at the same 'clock rate' as the dynamics of the world. This means that only *one* fresh sample is generated during the timescale over which things typically change in the outside dynamical world: we call this regime $n = 1$. Other work in Bayesian inference via MCMC considers matching the stationary distribution to the Bayesian posterior: this is the regime $n = \infty$.

Our line of investigation was inspired by empirical evidence that neural firing rates typically vary on order ten times faster than interesting features in the outside world. This led us to consider MCMC in the context of finite n, bridging the gap between the two regimes of prior work.

4.1 Summary of results

We assumed the brain wants to represent the Bayesian posterior distribution at some time through a sequence of n samples generated by a Markov chain. We identified two competing properties useful to the brain: first each sample taken individually should be approximately drawn from the true Bayesian posterior distribution. Second, successive samples should not be too dependent so that they well-describe the whole distribution. Our approach differs to prior work since we argue that the brain cares not only about the distribution of the n^{th} sample, but about the distribution of all n samples. We sought a cost function to encode these desired properties.

First we considered a cost function based on the KL divergence of the joint distribution of all n samples versus n independent copies of the true posterior distribution. We showed that this cost function preferred to decouple the Markov chain rather than get the samples to target the true posterior distribution, and thus does not encode our desired tradeoff.

We argued that the flaw in this cost function lay in the fact that it is not strictly necessary for the brain to have all consecutive samples highly independent. For the brain to make decisions based on the samples, it is more important that the n samples *taken together* provide a good

exploration of the target distribution. This can be encoded in the idea that the brain computes Monte Carlo estimates from the samples, and inspired a new cost function: the mean squared error of these Monte Carlo estimates.

We then asked, *which* Monte Carlo averages should we consider? We argued that the two most important quantities when making a decision under uncertainty are the average belief (the mean) and the degree of uncertainty therein (the variance). So we suggested minimising the mean squared error of all moments, but with a decaying weight for higher moments. We justified this heuristic by linking it to the KL divergence in the $n \to \infty$ limit using Edgeworth series.

We showed that our mean squared error cost function of decaying moments exactly encodes our desired tradeoff between hitting the target and independence of successive samples. We also showed that in the limit $n \to \infty$, the cost function purely favours matching the stationary distribution to the target, recovering the classical goal of Bayesian inference via MCMC.

This led us to another question: how might the brain *use* posterior samples? In particular, how should a downstream area of the brain extract information from the samples to make a decision, such as coordinating motor action. We showed that under certain conditions, more trust should be put in in the first and last samples in the series.

4.2 Predictions of the hypothesis

Sampling provides a mathematically appealing framework for the brain to represent uncertainty. Arguably the most important work in the near future should be in deriving experimentally testable predictions of the hypothesis. Based on our work we make two predictions.

The first prediction relates to what our mean squared error cost function says about decision-making and firing rate variability. Quick decisions must be based on small n, and in this regime variability in firing rate is problematic. Intuitively this is because decision-making approximated to first order should just use the mean of the distribution, but the mean is hard to estimate from a small number of samples that have high variance. On the other hand decisions over a longer timescale involve large n where variance is not problematic and in fact allows more complicated distributions to be encoded. This is described mathematically by our MSE cost function.

We could test for this effect by asking some subjects to make a decision in half a second, and others in 10 seconds. Since the subjects know in advance how long they have to make the decision, we could expect the neural dynamics to *respond* to the different timescales, and we should see higher firing rate variability during the longer decision. This is chiefly an effect of sampling, and should arguably be absent under the probabilistic population code framework.

The second prediction uses our result that, under certain conditions, decisions should be influenced more heavily by the first and last samples in the sequence of neural firing rates. This implies that under the sampling hypothesis, there should be a time-lagged correlation between the sampling neuron and the decision outcome (the lag should equal the time it takes to make the decision). It could be possible to search for such a time-lagged correlation using careful multielectrode array recordings, although this might be difficult amidst a sea of neural noise.

4.3 Implications for further work

We have developed a biologically motivated, normative means to draw n Markov samples that approximately target a Bayesian posterior distribution. Further theoretical work could try to use this framework to build online Bayesian inference algorithms in the sampling setting. Work could also go into asking more generally what constraints decision-making imposes on inference via sampling. Further experimental work could involve developing and testing our neural predictions in vivo.

Acknowledgements

I wish to thank Dr. Turner and Dr. Lengyel very warmly for all their help and advice throughout the course of this project. Thanks to the Cavendish Laboratory for organising the projects and giving so much flexibility. Figures were produced in Mathematica and using the Tikz package in LaTeX. All code is available on request.

Appendix A Short derivations

A.1 Stationary distribution of the linear Gaussian chain

Consider the Markov chain defined in Equation 2.1:

$$x_{i+1} = ax_i + \sigma\epsilon_i$$

with $0 < a < 1$ and $\epsilon_i \sim \mathcal{N}(0, 1)$. We are interested in the stationary distribution of the chain.

Neglecting the initial value of x_0, which will be damped away as $n \to \infty$, the chain is just a sum of Gaussian random variables. Therefore the stationary distribution is also Gaussian, and we just need to finds its mean and variance. The summands all have zero mean and therefore so does the stationary distribution. To find the variance we take the expected square of the equation $x_\infty = ax_\infty + \sigma\epsilon_\infty$. Finally this yields

$$\boxed{x_\infty \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-a^2}\right)} \tag{A.1}$$

A.2 Proof of the Markov decomposition of the KL divergence

Recall the factorisations given in Equations 3.1 and 3.2:

$$Q(\mathbf{x}) = q(x_1)\prod_{i=2}^{n} q(x_i|x_{i-1}) \qquad\qquad P(\mathbf{x}) = \prod_{i=1}^{n} p(x_i)$$

Let's use them to prove Equation 3.4:

$$\mathrm{KL}\left[Q(\mathbf{x})||P(\mathbf{x})\right] = \mathrm{KL}\left[q(x_1)||p(x_1)\right] + \sum_{i=2}^{n} \left\langle \mathrm{KL}\left[q(x_i|x_{i-1})||p(x_i)\right]\right\rangle_{q(x_{i-1})}$$

Proof.

$$\begin{aligned}
\mathrm{KL}\left[Q(\mathbf{x})||P(\mathbf{x})\right] &= \int d\mathbf{x}\, Q(\mathbf{x}) \log \frac{Q(\mathbf{x})}{P(\mathbf{x})} \\
&= \int d\mathbf{x}\, Q(\mathbf{x}) \log \frac{q(x_1)\prod_{i=2}^{n} q(x_i|x_{i-1})}{\prod_{i=1}^{n} p(x_i)} \\
&= \int d\mathbf{x}\, Q(\mathbf{x}) \left[\log \frac{q(x_1)}{p(x_1)} + \sum_{i=2}^{n} \log \frac{q(x_i|x_{i-1})}{p(x_i)}\right] \\
&= \int d\mathbf{x}\, q(x_1) \log \frac{q(x_1)}{p(x_1)} + \sum_{i=2}^{n} \int dx_i dx_{i-1}\, Q(x_i, x_{i-1}) \log \frac{q(x_i|x_{i-1})}{p(x_i)} \\
&= \mathrm{KL}\left[q(x_1)||p(x_1)\right] + \sum_{i=2}^{n} \left\langle \mathrm{KL}\left[q(x_i|x_{i-1})||p(x_i)\right]\right\rangle_{q(x_{i-1})}
\end{aligned}$$

□

A.3 Proof of the bias-variance decomposition of mean squared error

Consider an estimator $\hat{\lambda}$ of the quantity λ. We define the mean squared error of this estimator to be

$$MSE(\hat{\lambda}) = \left\langle (\hat{\lambda} - \lambda)^2 \right\rangle$$

We also define the estimator's bias, $b = \left\langle \hat{\lambda} \right\rangle - \lambda$, and variance, $V = \left\langle \hat{\lambda}^2 \right\rangle - \left\langle \hat{\lambda} \right\rangle^2$. Then the MSE can be broken down as

$$MSE(\hat{\lambda}) = b^2 + V$$

Proof.

$$\begin{aligned} b^2 + V &= \left(\left\langle \hat{\lambda} \right\rangle - \lambda \right)^2 + \left\langle \hat{\lambda}^2 \right\rangle - \left\langle \hat{\lambda} \right\rangle^2 \\ &= \left\langle \hat{\lambda} \right\rangle^2 - 2 \left\langle \hat{\lambda} \right\rangle \lambda + \lambda^2 + \left\langle \hat{\lambda}^2 \right\rangle - \left\langle \hat{\lambda} \right\rangle^2 \\ &= \lambda^2 - 2 \left\langle \hat{\lambda} \right\rangle \lambda + \left\langle \hat{\lambda}^2 \right\rangle \\ &= \left\langle (\hat{\lambda} - \lambda)^2 \right\rangle \\ &= MSE(\hat{\lambda}) \end{aligned}$$

□

Appendix B Applying the mean squared error cost function

We have the cost function

$$C = \sum_{k=1}^{\infty} \frac{1}{k!} MSE(\hat{\lambda}_k)$$

where $\hat{\lambda}_k$ is the Monte Carlo average of the k^{th} moment computed using n Markov chain samples. That is to say

$$\hat{\lambda}_k = \sum_{i=1}^{n} \frac{x_i^k}{n}$$

Defining the bias and variance of $\hat{\lambda}_k$ to be b_k and V_k respectively, we can rewrite this cost function using the bias-variance decomposition (Appendix A.3)

$$C = \sum_{k=1}^{\infty} \frac{b_k^2}{k!} + \sum_{k=1}^{\infty} \frac{V_k}{k!} \approx b_1^2 + \frac{1}{2} b_2^2 + V_1 + \frac{1}{2} V_2$$

We justify only keeping terms up to $k = 2$ by the fact that the factorial decay is so sharp.

Let's examine the consequences of this cost function for a specific example. Consider using the linear Gaussian chain ($x_{i+1} = ax_i + \sigma\epsilon_i$) to approximate a Gaussian with zero mean and variance τ^2.

Consider the case where the first sample is drawn from the chain's stationary distribution. This assumption greatly simplifies calculations, and is justified in the case when the world is changing so slowly that the neural sampler is effectively always at equilibrium. After some lengthy calculations, we can show that we have

$$b_1 = 0 \qquad\qquad b_2 = \frac{\sigma^2}{1-a^2} - \tau^2$$

$$V_1 = \frac{1}{n^2} \frac{\sigma^2}{1-a^2} \sum_{i,j=1}^{n} a^{|i-j|} \qquad\qquad V_2 = \frac{2}{n^2} \left(\frac{\sigma^2}{1-a^2} \right)^2 \sum_{i,j=1}^{n} a^{2|i-j|}$$

And therefore cost function

$$C = \frac{1}{2} \left[\frac{\sigma^2}{1-a^2} - \tau^2 \right]^2 + \frac{1}{n^2} \frac{\sigma^2}{1-a^2} \sum_{i,j=1}^{n} a^{|i-j|} + \frac{1}{n^2} \left(\frac{\sigma^2}{1-a^2} \right)^2 \sum_{i,j=1}^{n} a^{2|i-j|}$$

We note that the first term (a bias term) pressures the stationary distribution of the chain $\mathcal{N}(0, \frac{\sigma^2}{1-a^2})$ to hit the target distribution $\mathcal{N}(0, \tau^2)$. The other two terms (the variance terms) pressure a towards zero, i.e. decoupling the chain.

Finally we consider the limit $n \to \infty$. The only non-trivial terms are the variance terms of the form $\frac{1}{n^2} \sum_{i,j=1}^{n} a^{|i-j|}$. We note that these terms are bounded from above by $\frac{1}{n^2}[n + 2na + 2na^2 + ...]$. Summing the geometric progression, it is clear that these terms vanish as $n \to \infty$. Therefore the cost function reduces to

$$\lim_{n \to \infty} C = \frac{1}{2} \left[\frac{\sigma^2}{1-a^2} - \tau^2 \right]^2$$

and it is clear that in this limit, the cost function pressures the stationary distribution of the chain to exactly match the target distribution.

The behaviour is as follows:

1. for finite n, the cost function trades off each sample being drawn from the target distribution (reduces bias) against decoupling the chain (reduces variance)
2. for $n \to \infty$, the cost function makes the marginals hit the right target distribution, since variance is zero anyway.

Appendix C Relating KL divergence to cumulant discrepancies

We wish to write the KL divergence between q and p in terms of the discrepancies between their moments. We find it is easier and just as meaningful to write $\mathrm{KL}(q||p)$ in terms of the discrepancies between *cumulants*. We proceed by expressing p in terms of q and the cumulant discrepancies η_k, and then substituting this expression into the KL divergence.

C.1 Cumulant discrepancy cost function

Define the moment discrepancy cost function between $q(x)$ and $p(x)$ as

$$MD(q||p) = \sum_{k=1}^{\infty} \left(\left\langle x^k \right\rangle_q - \left\langle x^k \right\rangle_p \right)^2$$

A distribution can be specified precisely by its moments. Minimising this cost function sets all the moments equal to each other and hence exactly matches the distributions. What's more this cost function is easy to estimate from a sample (if we truncate it at some finite k). It is interesting to ask whether we can in general find some bound of $\mathrm{KL}(q||p)$ in terms of $MD(q||p)$.

First we note we can trade discussion of moments for cumulants as follows. We define

$$e^{H(t)} = \int q(x) e^{itx} dx$$

That is to say $H(t)$ is the logarithm of the Fourier transform of $q(x)$. Expanding $H(t)$ in powers of t yields

$$H(t) = \sum_{k=1}^{\infty} \kappa_k \frac{(it)^k}{k!}$$

where we define the cumulants $\{\kappa_k\}$.

Cumulants of a distribution $q(x)$ have some important properties [Kardar, 2007]:

i) κ_k can be written as a polynomial function of the first k moments

ii) $q(x)$ is uniquely defined by its cumulants

iii) κ_1 is the mean, κ_2 is the variance, κ_3 is the skewness and κ_4 is the excess kurtosis

iv) the Gaussian distribution has vanishing third and higher cumulants

Since κ_k is polynomial in the first k moments, we may as well construct a new cost function analogous to the moment discrepancy called the cumulant discrepancy:

$$\boxed{CD(q||p) = \sum_{k=1}^{\infty} (\kappa_k - \gamma_k)^2} \tag{C.1}$$

where κ_k and γ_k are the cumulants of q and p respectively. We can show that $CD(q||p) = 0$ if and only if $MD(q||p) = 0$.

Since there is a one-to-one mapping between a distribution $q(x)$ and its cumulants $\{\kappa_k\}$ we can think about the space of cumulants in some sense being dual to the space of probability density functions. Where the KL divergence acts on pdfs, the cumulant discrepancy is just the Euclidean distance between probability distributions in this dual space.

C.2 Writing $p(x)$ in terms of $q(x)$ and cumulant discrepancies

We define the k^{th} cumulant discrepancy between $q(x)$ and $p(x)$ to be $\eta_k = \gamma_k - \kappa_k$. Then we can write

$$CD(q||p) = \sum_{k=1}^{\infty} \eta_k^2$$

Our goal is now to find $p(x)$ in terms of $q(x)$ and the η_k, so that we can substitute this expression into $KL(q||p)$. First we note that using the inverse Fourier transform we can write

$$q(x) = \frac{1}{2\pi} \int e^{H(t)} e^{-itx} dt \tag{C.2}$$

where $H(t) = \sum_k \kappa_k \frac{(it)^k}{k!}$. The corresponding expression for p is then

$$p(x) = \frac{1}{2\pi} \int e^{H(t)} e^{-itx} \exp\left[\sum_k \eta_k \frac{(it)^k}{k!}\right] dt$$

By taking derivatives of Equation C.2 we can show

$$\frac{d^k q}{dx^k} = \frac{1}{2\pi} \int e^{H(t)} (-it)^k e^{-itx} dt$$

and substituting this into the expression for p (having expanded the exponential) yields

$$\boxed{p(x) = \exp\left[\sum_k \eta_k \frac{(-1)^k}{k!} \frac{d^k}{dx^k}\right] q(x)} \tag{C.3}$$

Here acting on $q(x)$ with a differential operator shifts it from a pdf with cumulants κ_k to another pdf with cumulants $\gamma_k = \kappa_k + \eta_k$. This can be viewed as the exponential map from the Lie algebra to the Lie group. Equation C.3 is well known in the context of the Gram-Charlier A series and the Edgeworth series, which are both methods of expanding arbitrary probability distributions around a Gaussian.

We can now, at least in principle, substitute Equation C.3 into the KL divergence to obtain an expression in the η_k. This is in general analytically intractable, and so we must settle for a series expansion in the η_k. In the next section we carry this out to leading order.

C.3 Series expansion of the KL divergence to leading order

Taking Equation C.3, writing the differential operator as $\frac{d}{dx} = D$, and expanding the exponential to first order in η_k we get

$$\begin{aligned} p &= \left[1 + \left[\sum_k \eta_k \frac{(-D)^k}{k!}\right]\right] q \\ &= q\left[1 + \frac{1}{q}\left[\sum_k \eta_k \frac{(-D)^k q}{k!}\right]\right] \end{aligned}$$

Taking logs yields

$$\log p = \log q + \frac{1}{q}\left[\sum_k \eta_k \frac{(-D)^k q}{k!}\right] + \frac{1}{2q^2}\left[\sum_k \eta_k \frac{(-D)^k q}{k!}\right]^2$$

Finally taking the KL divergence (and using the fact that $\int_{-\infty}^{\infty} \frac{d^k q}{dx^k} dx = 0$ for a well behaved probability distribution) we get to leading order in the η_k:

$$\boxed{KL(q||p) = \sum_{kl} \eta_k \eta_l \frac{(-1)^{k+l}}{2k!l!} \int dx \frac{1}{q} \frac{d^k q}{dx^k} \frac{d^l q}{dx^l}} \tag{C.4}$$

In principle we can evaluate this integral for a given q(x).

C.4 Special case: $q(x)$ is Gaussian

First consider the standard Normal distribution $q^*(x) = \frac{1}{\sqrt{2\pi}} \exp \frac{-x^2}{2}$. Taking derivatives it is easy to show that

$$\frac{d^k q^*}{dx^k} = (-1)^k H_k(x) q^*(x)$$

where $H_k(x)$ is the k^{th} Hermite polynomial. Hermite polynomials obey orthogonality relation

$$\frac{1}{k!} \int dx \, H_k(x) H_l(x) \frac{1}{\sqrt{2\pi}} \exp \frac{-x^2}{2} = \delta_{kl}$$

But we really want to consider the general Normal distribution $q(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(x-\mu)^2}{2\sigma^2}$. Changing variables yields the more general orthogonality relation for Gaussian $q(x)$

$$\frac{1}{k!} \int dx \, \frac{1}{q} \frac{d^k q}{dx^k} \frac{d^l q}{dx^l} = \left(\frac{-1}{\sigma} \right)^{k+l} \delta_{kl}$$

Plugging this into Equation C.4 yields (to leading order in η_k)

$$\boxed{KL(q||p) = \frac{1}{2} \sum_k \frac{\eta_k^2}{k! \sigma^{2k}}} \tag{C.5}$$

and the similarity to our cumulant discrepancy cost function $CD(q||p)$ in this case is clear.

C.5 Corollary

Using the fact that $\sum_k \frac{1}{k!} = e$, we can prove that if the cumulant discrepancies of $p(x)$ from a standard normal all satisfy $|\eta_k| \leq \epsilon$, then for small enough ϵ, $KL(q||p) \leq \frac{\epsilon^2 e}{2}$, where the bound is tight.

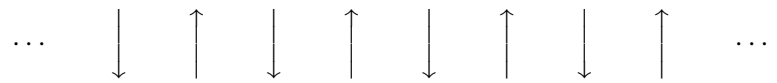
Appendix D Relevance of statistical physics to neuroscience

Neuroscience is a subject with very intimate links to statistical physics. These links stem from the fact that neural networks are large, complicated systems constructed from relatively simple building blocks: neurons and synapses. Statistical mechanics is exactly the tool to describe how macroscopic properties of systems composed of N building blocks emerge in the limit that $N \to \infty$.

Here we describe an example of the neuro/stat. mech. crossover: MCMC methods, which were used in the main text. We discuss one of their classic uses in computing properties of Ising models. We then describe how Ising models, or *spin glasses*, are relevant to neuroscience.

D.1 MCMC methods for Ising models

The 1D antiferromagnetic Ising model is a system of electrons (or 'spins') arranged on a line, and interacting such that neighbouring spins prefer to anti-align in the ground state.



Classically the spins are either up or down, so we can describe an arbitrary state of the system as a vector $\mathbf{x}$ with entries ± 1. The preference of spins to anti-align is encoded in Hamiltonian

$$\mathcal{H}(\mathbf{x}) = J \sum_{<ij>} x_i x_j \tag{D.1}$$

where $J > 0$ and the sum over $< ij >$ denotes summation over neighbouring lattice sites. All thermodynamic properties of the system can then be derived from partition function

$$\mathcal{Z} = \sum_{\mathbf{x}} \exp -\beta \mathcal{H}(\mathbf{x}) \tag{D.2}$$

The trouble is that if we generalise to arbitrary number of dimensions d, this summation is exponentially costly in d, and therefore generally intractable. But [MacKay, 2002, ch. 31] guides us to resolve the problem with Monte Carlo simulation via Gibbs sampling.

Starting from some arbitrary state, individual spins are sequentially updated to ± 1 with probability p. To find p, we treat the spin as a two level system sitting in an external field defined by the frozen state of the rest of the system. Following this Gibbs sampling scheme, the system provably relaxes to thermal equilibrium whatever the initial state. Thermodynamic quantities can then be estimated.

D.2 Spin glasses and associative memory

Computer memory operates in a different way to human memory. To access information on a computer hard disk you must provide a memory address, at which point the disk spins to that address location and retrieves whatever information is there. The brain, however, remembers via association: to jog your memory of an event, just a few details are enough. A partial memory triggers recall of the full memory. This is known as *content addressable* or *associative* memory.

[Little, 1974] and [Hopfield, 1982] noticed that Ising models provide a means to implement associative memory in the brain. The idea is this: we can view the ground state of the 1D antiferromagnet as encoding a binary vector $(1, 0, 1, 0, 1, ...)$. But generalising the Hamiltonian to $\mathcal{H}(\mathbf{x}) = \sum_{ij} J_{ij} x_i x_j$ for arbitrary J_{ij}, we can actually encode *arbitrary* binary strings into the ground state. This is known as a *spin glass* Hamiltonian, since the ground state is glassy – i.e.

exhibits a frozen disorder. These binary strings can be thought to encode memories. Cooling the system down to $T = 0$ will perfectly recover the memory.

For a non-frustrated nearest-neighbour Hamiltonian there is only one ground state (up to a global flip of spins). But seeing as we are now considering arbitrary J_{ij} (i.e. no longer nearest-neighbour) it is possible to have more than one ground state. In fact we can just superpose Hamiltonians corresponding to different binary strings, and this does not necessarily spoil the recall process. Indexing the different memories/binary strings by μ we can define the new superposed Hamiltonian

$$\mathcal{H}(\mathbf{x}) = \sum_{\mu} \sum_{ij} J_{ij}^{\mu} x_i x_j \tag{D.3}$$

A basin of attraction exists around each stored binary string within which the system's dynamics will fall into recalling that particular memory. This is what we mean when we say the memory is associative: starting the system in a state close to a particular memory recalls the whole thing.

[Amit et al., 1985] derived some nice statistical mechanical results about the capacity of such systems: that is, how many memories can be stored before the basins of attraction start to overlap and memory recall breaks down. These results were derived using replica theory, a technique described in [Altland and Simons, 2010].

The final step is to map Ising models back on to brain function. The simple idea is that a neuron either firing or not firing can be considered a binary event. Therefore the state of a neural network of n neurons, can be thought of as a binary vector of length n. The weight J_{ij} in the Hamiltonian is then just the synaptic strength between neurons i and j.

This is a beautiful idea, although it turns out that in practice the brain is far messier than this. Still the work of the physicists was important as together with neuroscientists they began to think of brain function in terms of the collective behaviour of dynamical systems, or as physicists say, *modes*.

References

[Aitchison and Lengyel, 2014] Aitchison, L. and Lengyel, M. (2014). The hamiltonian brain.

[Altland and Simons, 2010] Altland, A. and Simons, B. D. (2010). *Condensed Matter Field Theory*. Cambridge University Press, second edition. Cambridge Books Online.

[Amit et al., 1985] Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1985). Spin-glass models of neural networks. *Phys. Rev. A*, 32:1007–1018.

[Berkes et al., 2011] Berkes, P., Orbán, G., Lengyel, M., and Fiser, J. (2011). Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science*, 331:83–87.

[Borgwardt et al., 2006] Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., and Smola, A. J. (2006). Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57.

[Buesing et al., 2011] Buesing, L., Bill, J., Nessler, B., and Maass, W. (2011). Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons. *PLoS Computational Biology*, 7.

[Dayan and Abbott, 2005] Dayan, P. and Abbott, L. F. (2005). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press.

[Ernst, 2002] Ernst, M. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415(6870):429–433. cited By (since 1996) 622.

[Fiser et al., 2010] Fiser, J., Berkes, B., Orbán, G., and Lengyel, M. (2010). Statistically optimal perception and learning: from behavior to neural representations. *Trends Cogn Sci*, 14:119–130.

[Gershman et al., 2012] Gershman, S., Vul, E., and Tenenbaum, J. B. (2012). Multistability and perceptual inference. *Neural Computation*, 24:1–24.

[Greaves-Tunnell, 2015] Greaves-Tunnell, A. (2015). An optimization perspective on approximate neural filtering. Master's thesis, University of Cambridge.

[Haefner et al., 2016] Haefner, R. M., Berkes, P., and Fiser, J. (2016). Perceptual decision-making as probabilistic inference by neural sampling. *Neuron*.

[Hennequin et al., 2014] Hennequin, G., Aitchison, L., and Lengyel, M. (2014). Fast sampling-based inference in balanced neuronal networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 27*, pages 2240–2248. Curran Associates, Inc.

[Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558.

[Hoyer and Hyvärinen, 2003] Hoyer, P. O. and Hyvärinen, A. (2003). Interpreting neural response variability as monte carlo sampling of the posterior. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 293–300. MIT Press.

[Kardar, 2007] Kardar, M. (2007). *Statistical Physics of Particles*. Cambridge University Press.

[Kording and Wolpert, 2004] Kording, K. P. and Wolpert, D. M. (2004). Bayesian integration in sensorimotor learning. *Nature*, 427(6971):244–247.

[Kutschireiter et al., 2015] Kutschireiter, A., Surace, S. C., Sprekeler, H., and Pfister, J.-P. (2015). A neural implementation for nonlinear filtering. *http://arxiv.org/abs/1508.06818*.

[Lieder et al., 2012] Lieder, F., Griffiths, T. L., and Goodman, N. D. (2012). Burn-in, bias, and the rationality of anchoring. In *NIPS*.

[Little, 1974] Little, W. (1974). The existence of persistent states in the brain. *Mathematical Biosciences*, 19(1):101 – 120.

[Ma et al., 2006] Ma, W. J., Beck, J. M., Latham, P. E., and Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nat Neurosci*, 9(11):1432–1438.

[MacKay, 2002] MacKay, D. J. C. (2002). *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA.

[Murray et al., 2014] Murray, J. D., Bernacchia, A., Freedman, D. J., Romo, R., Wallis, J. D., Cai, X., Padoa-Schioppa, C., Pasternak, T., Seo, H., Lee, D., and Wang, X.-J. (2014). A hierarchy of intrinsic timescales across primate cortex. *Nat Neurosci*, 17(12):1661–1663.

[Pouget et al., 2003] Pouget, A., Dayan, P., and Zemel, R. S. (2003). Inference and computation with population codes. *Annual Review of Neuroscience*, 26(1):381–410. PMID: 12704222.

[Salimans et al., 2015] Salimans, T., Kingma, D. P., and Welling, M. (2015). Markov chain monte carlo and variational inference: Bridging the gap. In *ICML*.

[Spira and Hai, 2013] Spira, M. E. and Hai, A. (2013). Multi-electrode array technologies for neuroscience and cardiology. *Nat Nano*, 8(2):83–94.

[Turner and Sahani, 2011] Turner, R. E. and Sahani, M. (2011). *Bayesian Time Series Models*, chapter "Two problems with variational expectation maximisation for time series models". Cambridge University Press.

[Vul et al., 2014] Vul, E., Goodman, N., Griffiths, T. L., and Tenenbaum, J. B. (2014). One and done? optimal decisions from very few samples. *Cognitive Science*, 38(4):599–637.

[Worrall, 2014] Worrall, D. E. (2014). The neural sampling hypothesis in dynamic environments. Master's thesis, University of Cambridge.