Lecture 5: Approximate inference

Samuel Gershman

Harvard University

Roadmap

- Exact inference is typically intractable. Can the brain get close to the right answer?

Roadmap

- Exact inference is typically intractable. Can the brain get close to the right answer?
- Sampling approximations harness randomness, achieving asymptotic correctness in the limit of many samples. Offers a functional explanation for the ubiquity of noise in the brain.

Roadmap

- Exact inference is typically intractable. Can the brain get close to the right answer?
- Sampling approximations harness randomness, achieving asymptotic correctness in the limit of many samples. Offers a functional explanation for the ubiquity of noise in the brain.
- Variational approximations replace complex posteriors with simpler parametric forms, converting inference into a more tractable optimization problem (minimizing free energy). Can be implemented using a hierarchical architecture in which feedback signals convey predictions and feedforward signals convey prediction errors.

Contour detection example

- Look for the light switch in the dark; brain's job is to identify contours that look like light switches.

Contour detection example

- Look for the light switch in the dark; brain's job is to identify contours that look like light switches.
- Contour is a collection of adjacent edges that vary smoothly over space.

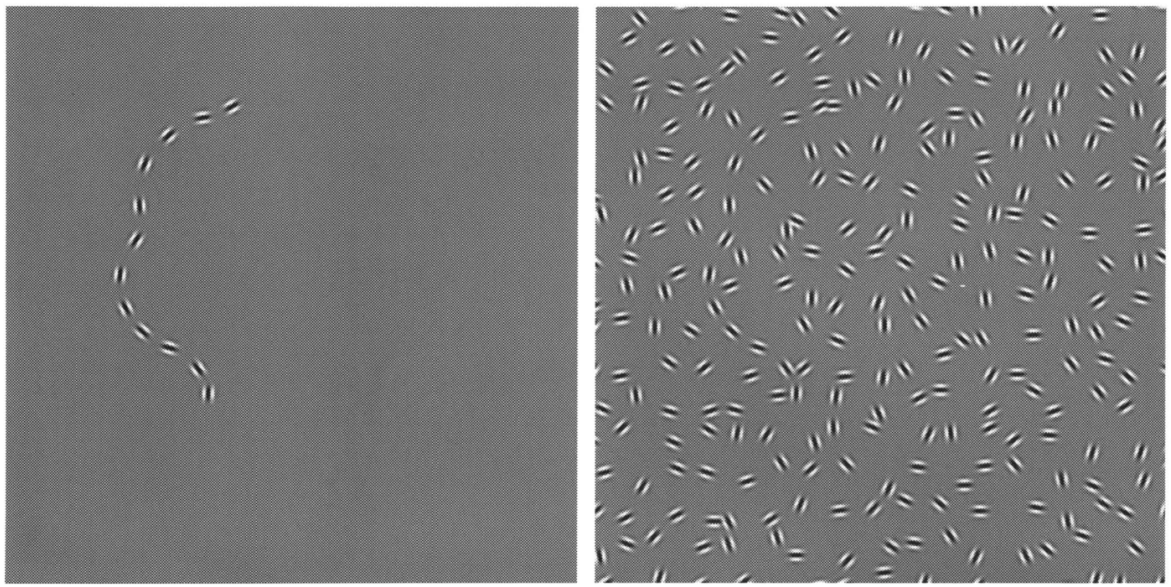
Contour detection example

- Look for the light switch in the dark; brain's job is to identify contours that look like light switches.
- Contour is a collection of adjacent edges that vary smoothly over space.
- The inference problem is high-dimensional: need to infer the edge orientation at each point in space (the orientation field).

Contour detection example

- Look for the light switch in the dark; brain's job is to identify contours that look like light switches.
- Contour is a collection of adjacent edges that vary smoothly over space.
- The inference problem is high-dimensional: need to infer the edge orientation at each point in space (the orientation field).
- Can't be parallelized (edge orientations covary).

Contour detection example



[Field, 1993]

What makes inference hard

- Recall our problem: compute the posterior $p(s|x) = \frac{p(x|s)p(s)}{\sum_{s'} p(x|s')p(s')}$ over hidden state s given data x.

What makes inference hard

- Recall our problem: compute the posterior $p(s|x) = \frac{p(x|s)p(s)}{\sum_{s'} p(x|s')p(s')}$ over hidden state s given data x.
- In the contour detection example, x is an image and s is the orientation at each location in the image.

What makes inference hard

- Recall our problem: compute the posterior $p(s|x) = \frac{p(x|s)p(s)}{\sum_{s'} p(x|s')p(s')}$ over hidden state s given data x.
- In the contour detection example, x is an image and s is the orientation at each location in the image.
- Hard part is the denominator (the marginal likelihood), which involves summing over all possible states. In the contour detection example, this is the space of all possible orientation fields (exponential in the number of locations).

Sampling approximations

- Key idea: replace exhaustive enumeration of the hidden states with a set of K samples, $\{s^1, \ldots, s^K\}$, drawn from $p(s|x)$:

$$p(s|x) \approx \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}[s^k = s],$$

where $\mathbb{I}[\cdot] = 1$ if its argument is true, and 0 otherwise.

Sampling approximations

- Key idea: replace exhaustive enumeration of the hidden states with a set of K samples, $\{s^1, \ldots, s^K\}$, drawn from $p(s|x)$:

$$p(s|x) \approx \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}[s^k = s],$$

  where $\mathbb{I}[\cdot] = 1$ if its argument is true, and 0 otherwise.
- The probability of s is approximately equal to the proportion of samples with that value.

Sampling approximations

- Key idea: replace exhaustive enumeration of the hidden states with a set of K samples, $\{s^1, \ldots, s^K\}$, drawn from $p(s|x)$:

$$p(s|x) \approx \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}[s^k = s],$$

  where $\mathbb{I}[\cdot] = 1$ if its argument is true, and 0 otherwise.
- The probability of s is approximately equal to the proportion of samples with that value.
- As K gets larger, the approximation gets increasingly accurate.

Markov chain Monte Carlo

- Generating samples from the posterior is non-trivial.

Markov chain Monte Carlo

- Generating samples from the posterior is non-trivial.
- One strategy is to generate samples from a dynamical system that converges to the posterior.

Markov chain Monte Carlo

- Generating samples from the posterior is non-trivial.
- One strategy is to generate samples from a dynamical system that converges to the posterior.
- When the dynamical system is characterized by a transition probability $T(s'|s)$ that doesn't depend on any previous samples, it is called a *Markov chain*, and the sampling algorithm is a form of *Markov chain Monte Carlo* (MCMC).

The Metropolis-Hastings algorithm

- Factor the transition probability into a *proposal distribution* $G(s'|s)$ and an *acceptance distribution* $A(s'|s)$:

$$T(s'|s) = G(s'|s, x)A(s'|s)$$

The Metropolis-Hastings algorithm

- Factor the transition probability into a *proposal distribution* $G(s'|s)$ and an *acceptance distribution* $A(s'|s)$:

$$T(s'|s) = G(s'|s,x)A(s'|s)$$

- The proposal distribution can be arbitrary. The acceptance distribution is given by:

$$A(s'|s) = \min\left[1, \frac{p(s',x)G(s|s',x)}{p(s,x)G(s'|s,x)}\right]$$

The Metropolis-Hastings algorithm

- Factor the transition probability into a *proposal distribution* $G(s'|s)$ and an *acceptance distribution* $A(s'|s)$:

$$T(s'|s) = G(s'|s,x)A(s'|s)$$

- The proposal distribution can be arbitrary. The acceptance distribution is given by:

$$A(s'|s) = \min\left[1, \frac{p(s',x)G(s|s',x)}{p(s,x)G(s'|s,x)}\right]$$

- Proposals are more likely to be accepted when they increase probability under $p(s,x)$. The G ratio corrects for sampling bias.

Gibbs sampling

- Special case of Metropolis-Hastings, where proposals are sampled from conditional distributions:

$$G(s'|s,x) = \sum_n p(n)p(s'_n|s_{/n},x)$$

where $s_{/n}$ is the set of "sites" excluding site s_n, and $p(n)$ is the probability of modification at site n (typically uniform).

Gibbs sampling

- Special case of Metropolis-Hastings, where proposals are sampled from conditional distributions:

$$G(s'|s,x) = \sum_n p(n)p(s'_n|s_{/n},x)$$

  where $s_{/n}$ is the set of "sites" excluding site s_n, and $p(n)$ is the probability of modification at site n (typically uniform).
- Acceptance probability is always 1.

Application to contour detection

- Let $s_n \in [0, 2\pi]$ denote the orientation at location n. The state vector s is the orientation field.

Application to contour detection

- Let $s_n \in [0, 2\pi]$ denote the orientation at location n. The state vector s is the orientation field.
- In natural images, edges at nearby locations tend to have similar orientations, which motivates a smoothness prior:

$$p(s) \propto \exp\left[\sum_n \sum_m H_{nm} \cos(s_n - s_m)\right]$$

where $H_{nm} > 0$ if locations n and m are neighbors (0 otherwise).

Application to contour detection

- Let $s_n \in [0, 2\pi]$ denote the orientation at location n. The state vector s is the orientation field.
- In natural images, edges at nearby locations tend to have similar orientations, which motivates a smoothness prior:

$$p(s) \propto \exp\left[\sum_n \sum_m H_{nm} \cos(s_n - s_m)\right]$$

  where $H_{nm} > 0$ if locations n and m are neighbors (0 otherwise).
- Sensory data $x = (x_1, \ldots, x_D)$ are the spike counts of Poisson neurons with tuning curves $\{f_d(s)\}$.

Application to contour detection

- Spatially localized cosine tuning functions:

$$f_d(s_n) = \exp\left[\frac{1}{\nu}\cos(s_n - s^*_{dn})\right]$$

where s^*_{dn} is the preferred orientation for neuron d at location n, and ν is the tuning width.

Application to contour detection

- Spatially localized cosine tuning functions:

$$f_d(s_n) = \exp\left[\frac{1}{\nu}\cos(s_n - s^*_{dn})\right]$$

where s^*_{dn} is the preferred orientation for neuron d at location n, and ν is the tuning width.

- Under a Poisson distribution, we get the following likelihood:

$$p(x|s) \propto \exp\left[\frac{1}{\nu}\sum_d x_d \sum_n \cos(s_n - s^*_{dn})\right]$$

where we have again made use of the assumption that $\sum_d f_d(s)$ is a constant (satisfied if the tuning functions are shifted copies of one another and tile the state space).

Application to contour detection

- Putting it all together with Bayes' rule:

$$p(s|x) \propto \exp\left[\frac{1}{\nu}\sum_d x_d \sum_n \cos(s_n - s^*_{dn}) + \sum_{n,m} H_{nm}\cos(s_n - s_m)\right]$$

Application to contour detection

- Putting it all together with Bayes' rule:

$$p(s|x) \propto \exp\left[\frac{1}{\nu}\sum_d x_d \sum_n \cos(s_n - s^*_{dn}) + \sum_{n,m} H_{nm}\cos(s_n - s_m)\right]$$

- Gibbs sampling can be implemented by iterating over locations and choosing new orientations conditional on the orientations at the other locations:

$$p(s_n|s_{/n}, x) \propto \exp\left[\frac{1}{\nu}\sum_d x_d \cos(s_n - s^*_{dn}) + \sum_m H_{nm}\cos(s_n - s_m)\right]$$

where d sums over all input neurons tuned to location n.

Application to contour detection

- If we discretize the orientations into $\{\tilde{s}_j\}$, we can calculate the normalizing constant of the conditional distribution and tractably sample from it.

Application to contour detection

- If we discretize the orientations into $\{\tilde{s}_j\}$, we can calculate the normalizing constant of the conditional distribution and tractably sample from it.
- This is just another example of the softmax equation.

Mapping to a neural circuit

- Let $z_d(t) \in \{0, 1\}$ denote the spike train of input neuron d.

Mapping to a neural circuit

- Let $z_d(t) \in \{0, 1\}$ denote the spike train of input neuron d.
- Output neurons are indexed both by location (n) and orientation (j). Each output neuron integrates input spikes linearly, along with "lateral" contributions from other output neurons whose spike trains are presented by $\{y_{mj}(t)\}$:

$$I_{nj}(t) = \frac{1}{\nu} \sum_d z_d(t) \cos(s_n - s^*_{dn}) + \sum_m H_{nm} \cos(s_n - s_m) y_{mj}(t)$$

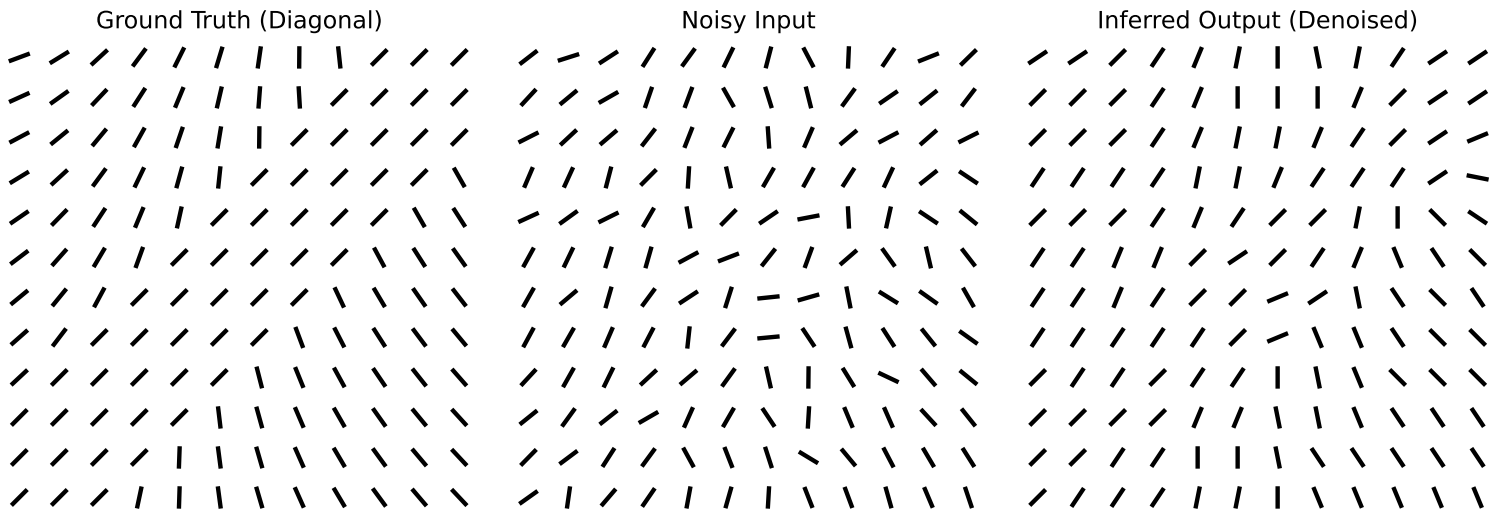
Mapping to a neural circuit

- Let $z_d(t) \in \{0, 1\}$ denote the spike train of input neuron d.
- Output neurons are indexed both by location (n) and orientation (j). Each output neuron integrates input spikes linearly, along with "lateral" contributions from other output neurons whose spike trains are presented by $\{y_{mj}(t)\}$:

$$I_{nj}(t) = \frac{1}{\nu} \sum_d z_d(t) \cos(s_n - s^*_{dn}) + \sum_m H_{nm} \cos(s_n - s_m) y_{mj}(t)$$

- Membrane potential $\mu_{nj}(t)$ is a perfect integrator, $C\dot{\mu}_{nj}(t) = I_{nj}(t)$, that is exponentiated to produce the intensity function (expected firing rate of a Poisson process): $\rho_{nj}(t) \propto \exp[\mu_{nj}(t)]$. Feedback inhibition reflects total activity of neurons tuned to the same location but different orientations.

Demonstration



Summary so far

- We can view this circuit as implementing a stochastic dynamical system that converges to the posterior; spikes of the output neurons are posterior samples.

Summary so far

- We can view this circuit as implementing a stochastic dynamical system that converges to the posterior; spikes of the output neurons are posterior samples.
- Sampling models offer a fundamentally different point of view—noise is a feature, not a bug.

Summary so far

- We can view this circuit as implementing a stochastic dynamical system that converges to the posterior; spikes of the output neurons are posterior samples.
- Sampling models offer a fundamentally different point of view—noise is a feature, not a bug.
- Can we find evidence for neural sampling in the brain?

Neural evidence for sampling

- Spontaneous neural activity prior to stimulus onset should reflect samples from the prior, whereas stimulus-evoked activity should reflect samples from the posterior.

Neural evidence for sampling

- Spontaneous neural activity prior to stimulus onset should reflect samples from the prior, whereas stimulus-evoked activity should reflect samples from the posterior.
- Generally, the posterior will be narrower (lower variance) than the prior.

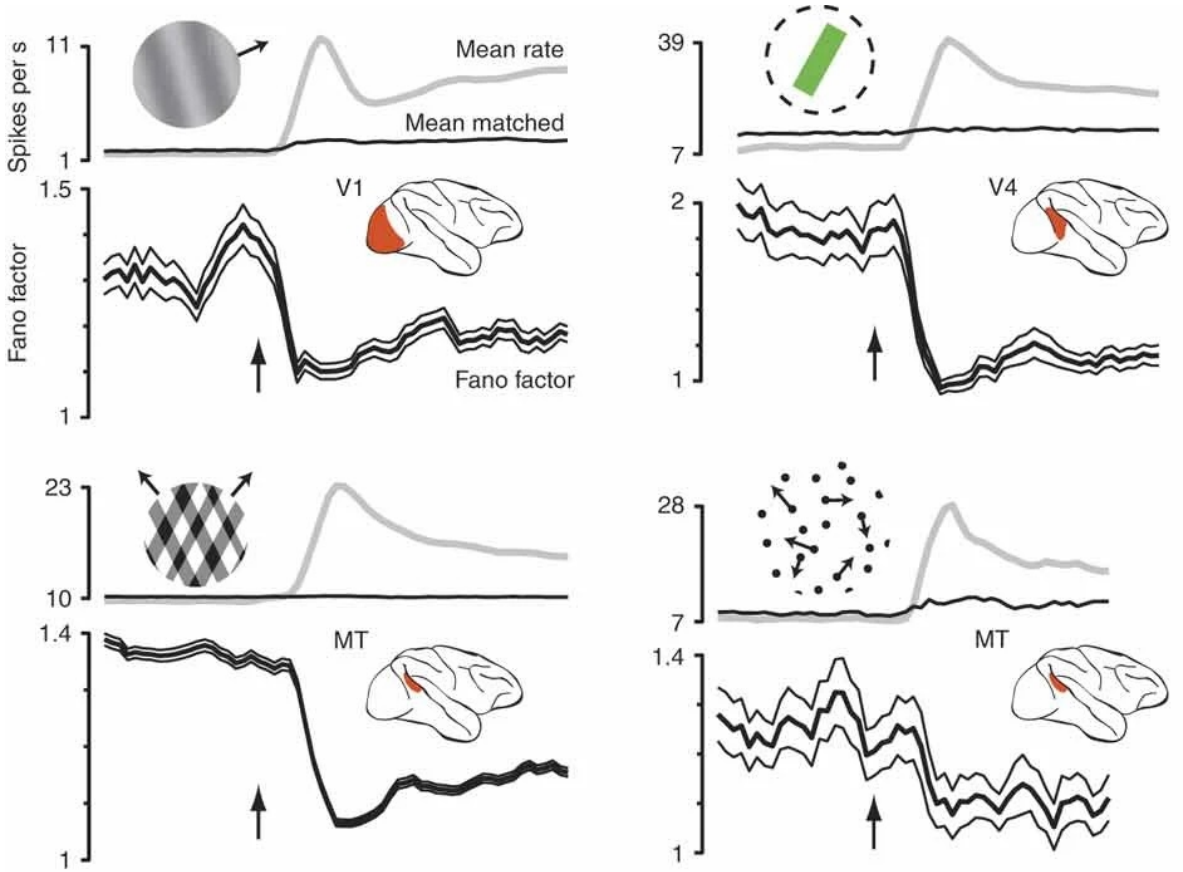
Neural evidence for sampling

- Spontaneous neural activity prior to stimulus onset should reflect samples from the prior, whereas stimulus-evoked activity should reflect samples from the posterior.
- Generally, the posterior will be narrower (lower variance) than the prior.
- Under the sampling hypothesis, neural variability should increase monotonically with posterior variance. Intuitively, this is because when variance is high the samples must explore a broader range of states.

Neural evidence for sampling

- Spontaneous neural activity prior to stimulus onset should reflect samples from the prior, whereas stimulus-evoked activity should reflect samples from the posterior.
- Generally, the posterior will be narrower (lower variance) than the prior.
- Under the sampling hypothesis, neural variability should increase monotonically with posterior variance. Intuitively, this is because when variance is high the samples must explore a broader range of states.
- This implies that stimulus onset should "quench" neural variability, as observed experimentally.

Stimulus onset quenches neural variability



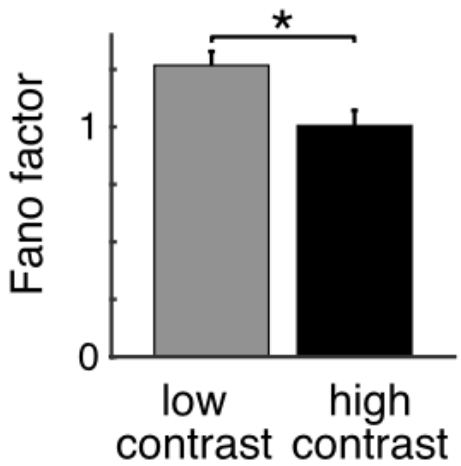
[Churchland et al 2010]
Fano factor: ratio of variance to mean.

Neural evidence for sampling

- Increasing uncertainty (e.g., by reducing stimulus contrast) should increase neural variability.

Neural evidence for sampling

- Increasing uncertainty (e.g., by reducing stimulus contrast) should increase neural variability.
- Consistent with this prediction, neural variability in primary visual cortex (V1) is higher for low contrast stimuli.



[Orban et al 2016; data from Ecker et al 2010]

Neural evidence for sampling

- Stimulus-evoked activity, when averaged across trials, should resemble spontaneous activity. This is because the expected posterior is the prior:

$$\mathbb{E}[p(s|x)] = \sum_x p(x)p(s|x) = p(s)$$

Neural evidence for sampling

- Stimulus-evoked activity, when averaged across trials, should resemble spontaneous activity. This is because the expected posterior is the prior:

$$\mathbb{E}[p(s|x)] = \sum_x p(x)p(s|x) = p(s)$$

- Critically, this is only true if the stimulus distribution $p(x)$ reflects the "natural statistics" of stimuli in the real world, assuming that the brain's prior, $p(s)$, is adapted to these statistics.

Neural evidence for sampling

- Stimulus-evoked activity, when averaged across trials, should resemble spontaneous activity. This is because the expected posterior is the prior:

$$\mathbb{E}[p(s|x)] = \sum_x p(x)p(s|x) = p(s)$$

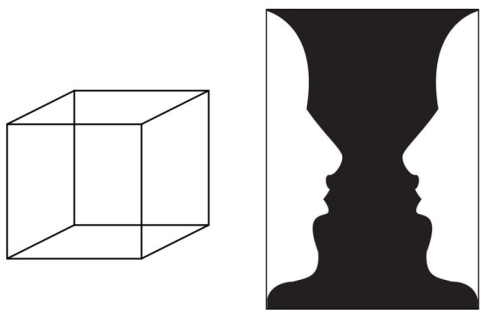
- Critically, this is only true if the stimulus distribution $p(x)$ reflects the "natural statistics" of stimuli in the real world, assuming that the brain's prior, $p(s)$, is adapted to these statistics.
- The distribution of spontaneous activity was more similar to the marginal distribution of stimulus-evoked activity for natural images compared to artificial images [Orban et al 2016].

Behavioral evidence for sampling

- Perceptual multistability arises when the brain switches repeatedly (and usually stochastically) between different interpretations of the same visual input.

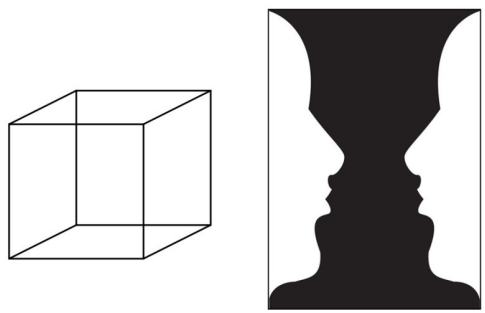
Behavioral evidence for sampling

- Perceptual multistability arises when the brain switches repeatedly (and usually stochastically) between different interpretations of the same visual input.
- Ambiguous images like the Necker cube and the face-vase illusion can be interpreted in different ways by the same observer over a short interval of time.



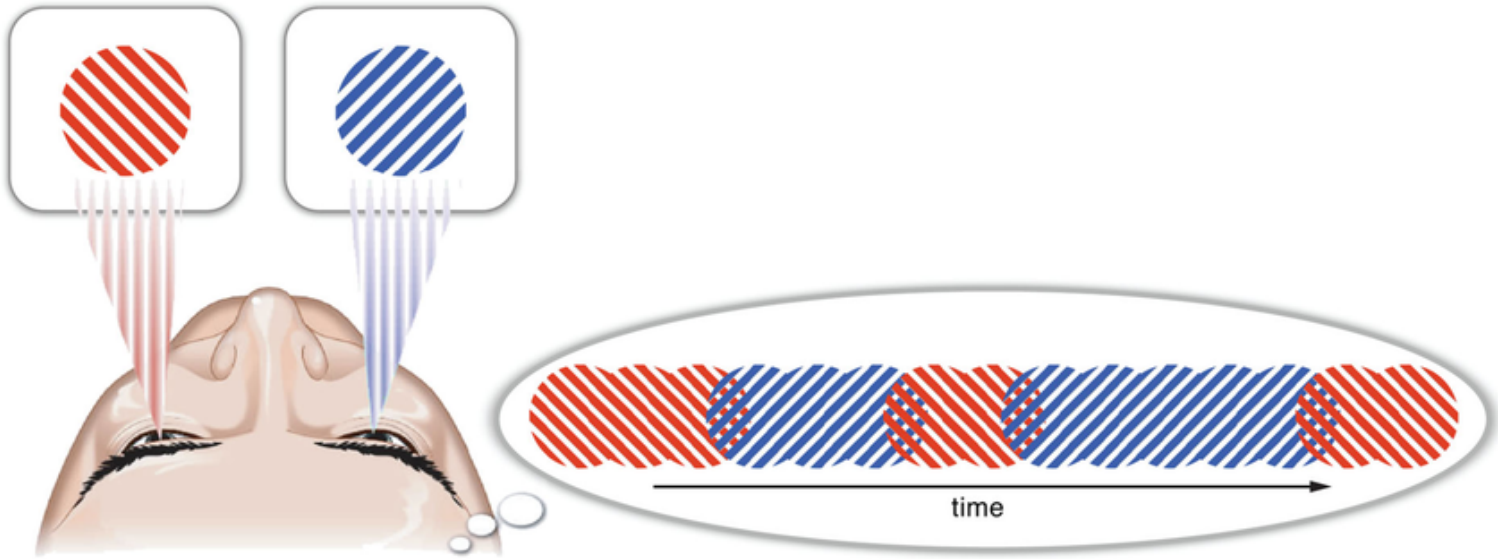
Behavioral evidence for sampling

- Perceptual multistability arises when the brain switches repeatedly (and usually stochastically) between different interpretations of the same visual input.
- Ambiguous images like the Necker cube and the face-vase illusion can be interpreted in different ways by the same observer over a short interval of time.

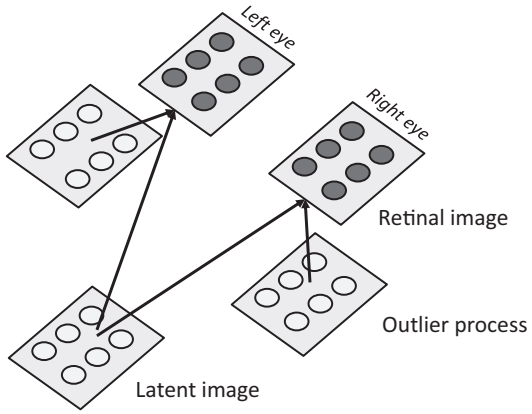


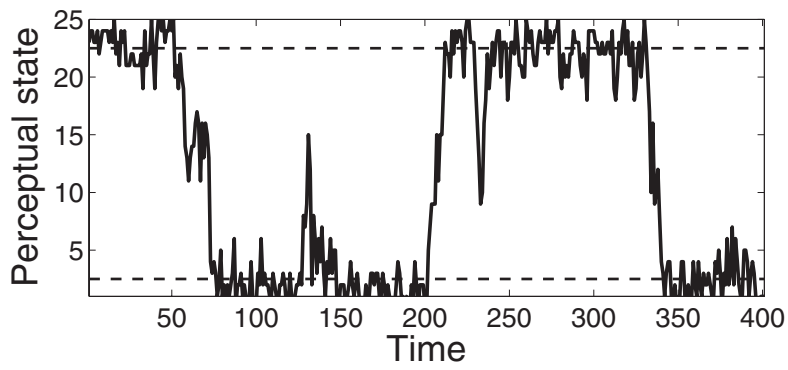
- Another example is binocular rivalry: when different images are presented to each eye, typically only one image is perceived at a time, with stochastic switches between the dominant image.

Binocular rivalry



A probabilistic model of binocular rivalry





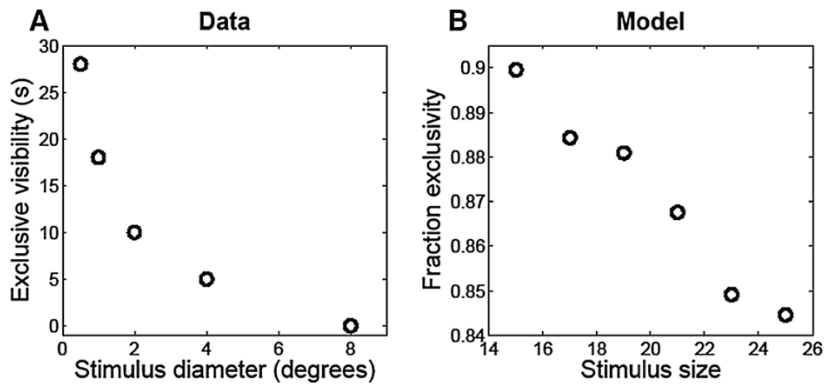
[Gershman et al 2012]

Piecemeal rivalry

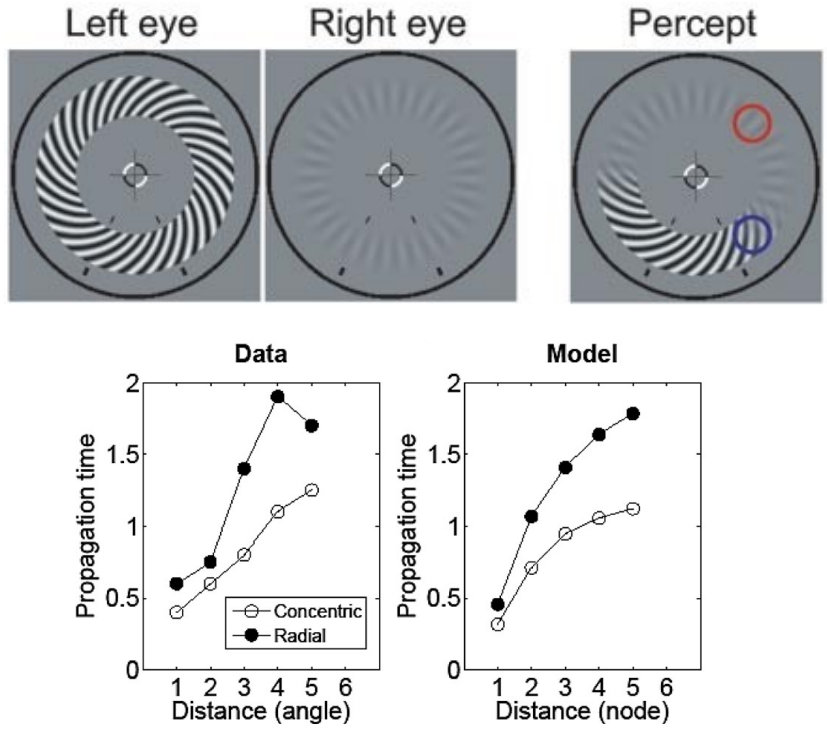
- Switches are not always all-or-none: large images produce "piecemeal" switches, where one part of the image switches before other parts [O'Shea et al 1997].

Piecemeal rivalry

- Switches are not always all-or-none: large images produce "piecemeal" switches, where one part of the image switches before other parts [O'Shea et al 1997].
- This makes sense if sampling is operating at the level of image parts. For larger images, the dependencies between different parts become weaker, allowing piecemeal switches.



Traveling waves in binocular rivalry



[Wilson et al 2001; Lee et al 2005; Gershman et al 2012]

Fusion in binocular rivalry

- Fusion is more likely when both images are low contrast [Burke et al 1999] and when they are similar [Knapen et al 2007].

Fusion in binocular rivalry

- Fusion is more likely when both images are low contrast [Burke et al 1999] and when they are similar [Knapen et al 2007].
- Under the sampling hypothesis, fusion arises when the two posterior modes are not well-separated, either because their variances are large (low contrast) or because their modes are near one another (high similarity). In this case, sampling doesn't bounce as much between the two modes, but instead spends more time in the high density area between them.

Variational approximations

- MCMC algorithms are asymptotically correct: if you run them long enough, you'll approximate the posterior to an arbitrary degree of precision.

Variational approximations

- MCMC algorithms are asymptotically correct: if you run them long enough, you'll approximate the posterior to an arbitrary degree of precision.
- However, you might need to run them a long time if the problem is complex.

Variational approximations

- MCMC algorithms are asymptotically correct: if you run them long enough, you'll approximate the posterior to an arbitrary degree of precision.
- However, you might need to run them a long time if the problem is complex.
- Alternative: use an approximation algorithm that produces an answer more quickly, but doesn't enjoy asymptotic correctness. Variational approximations offer a general framework for doing this.

Free energy minimization

- Basic idea: turn inference into a constrained optimization problem.

Free energy minimization

- Basic idea: turn inference into a constrained optimization problem.
- Goal: find an approximate posterior $q \in \mathcal{Q}$ that gets closest to the posterior, where $\mathcal{Q}$ is chosen in such a way that both finding and evaluating q is relatively fast.

Free energy minimization

- Basic idea: turn inference into a constrained optimization problem.
- Goal: find an approximate posterior $q \in \mathcal{Q}$ that gets closest to the posterior, where $\mathcal{Q}$ is chosen in such a way that both finding and evaluating q is relatively fast.
- Optimization problem:

$$q^* = \operatorname*{argmin}_{q \in \mathcal{Q}} \mathcal{F}[q(s|x)]$$

where $\mathcal{F}[q(s|x)] = \sum_s q(s|x) \log \frac{q(s|x)}{p(x,s)}$ is the *variational free energy*.

Free energy minimization

- Basic idea: turn inference into a constrained optimization problem.
- Goal: find an approximate posterior $q \in \mathcal{Q}$ that gets closest to the posterior, where $\mathcal{Q}$ is chosen in such a way that both finding and evaluating q is relatively fast.
- Optimization problem:

$$q^* = \operatorname*{argmin}_{q \in \mathcal{Q}} \mathcal{F}[q(s|x)]$$

  where $\mathcal{F}[q(s|x)] = \sum_s q(s|x) \log \frac{q(s|x)}{p(x,s)}$ is the *variational free energy*.
- Equivalent to minimizing KL divergence between $q(s|x)$ and $p(s|x)$.

The Laplace approximation

- Restrict $\mathcal{Q}$ to the set of Gaussian posteriors: $q(s|x) = \mathcal{N}(s; \hat{s}, \Sigma)$, where the mean and covariance are variational parameters.

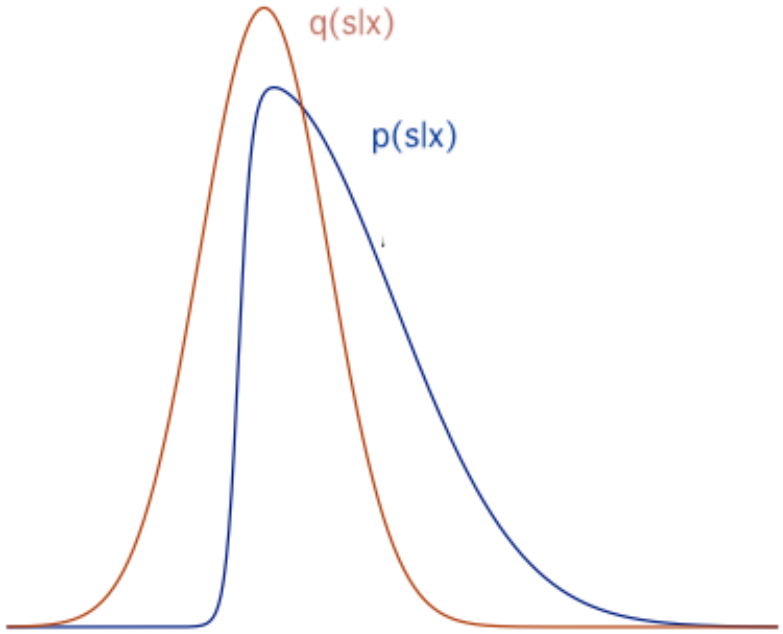
The Laplace approximation

- Restrict $\mathcal{Q}$ to the set of Gaussian posteriors: $q(s|x) = \mathcal{N}(s; \hat{s}, \Sigma)$, where the mean and covariance are variational parameters.
- We can obtain a tractable approximation of the free energy if we linearize $\log p(x, s)$ with a second-order Taylor series expansion around $\hat{s}$:

$$\log p(x, s) \approx \log p(x, \hat{s}) + (s - \hat{s})^\top \nabla_s \log p(x, \hat{s}) - \frac{1}{2}(s - \hat{s})^\top \Lambda (s - \hat{s})$$

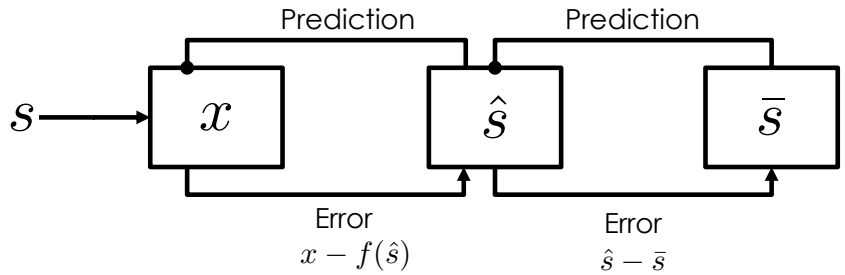
where $\Lambda = -\nabla_s \nabla_s \log p(x, \hat{s})$ is the Hessian (matrix of 2nd derivatives) of the negative log likelihood evaluated at $\hat{s}$.

The Laplace approximation



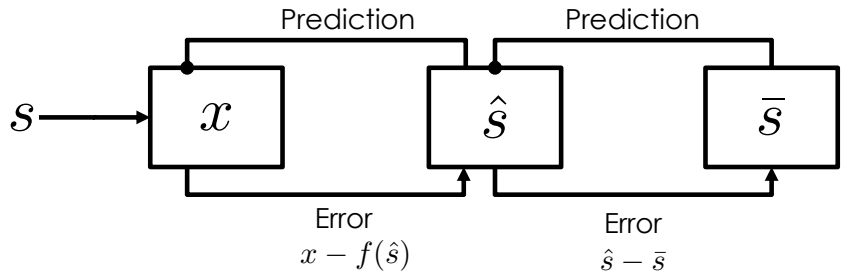
Predictive coding

- Under the Gaussian variational family and the Laplace approximation, we can derive a hierarchical architecture for inference.



Predictive coding

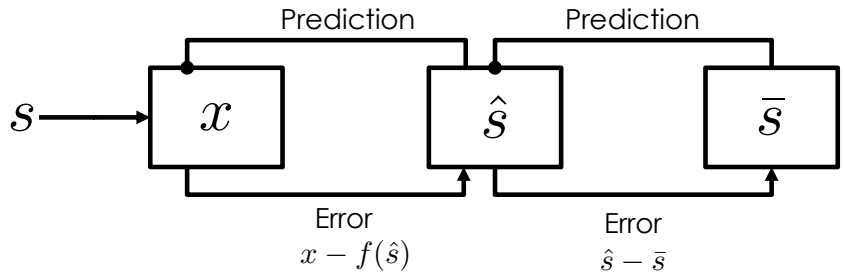
- Under the Gaussian variational family and the Laplace approximation, we can derive a hierarchical architecture for inference.



- "Prediction" neurons $y(t)$ report the inferred state, $\hat{s}$. They receive input from "error" neurons reporting the difference between observed and expected signals.

Predictive coding

- Under the Gaussian variational family and the Laplace approximation, we can derive a hierarchical architecture for inference.



- "Prediction" neurons $y(t)$ report the inferred state, $\hat{s}$. They receive input from "error" neurons reporting the difference between observed and expected signals.
- Two kinds of error neurons: "bottom-up" error neurons report the difference between sensory signals $z(t)$ and the expected firing rate under the inferred state, $f(\hat{s})$; "top-down" error neurons report the difference between the inferred state and the expected state under the prior distribution, $\bar{s}$.

Evidence for predictive coding

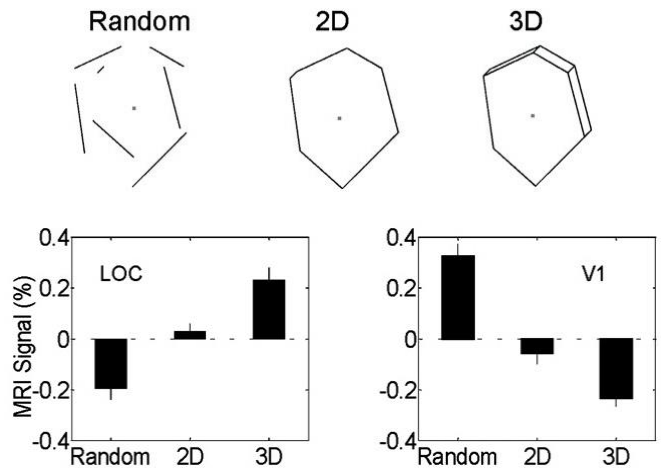
- Predictive coding suggests that predictions should suppress lower-level error signals.

Evidence for predictive coding

- Predictive coding suggests that predictions should suppress lower-level error signals.
- V1 neurons in layers 2/3 (the feedforward pathway thought to convey errors) fire more when novel images are presented, and these novelty responses decrease as the images are repeatedly presented [Homan et al 2022].

Evidence for predictive coding

- Predictive coding suggests that predictions should suppress lower-level error signals.
- V1 neurons in layers 2/3 (the feedforward pathway thought to convey errors) fire more when novel images are presented, and these novelty responses decrease as the images are repeatedly presented [Homan et al 2022].
- Higher-level visual areas respond more to images with coherent shape structure, accompanied by *decreases* in the responses of lower-level regions.



Study question

What are the complementary strengths and weaknesses of sampling vs. variational approximations? How might the brain decide which to deploy in a given context?

Summary

- The brain has multiple biologically plausible options for approximate inference. These are not mutually exclusive.

Summary

- The brain has multiple biologically plausible options for approximate inference. These are not mutually exclusive.
- Different algorithms could be used by different parts of the brain, based on their complementary strengths and weaknesses for different tasks. For tasks requiring fast sensory processing, it may make sense to rely on primarily feedforward algorithms, whereas for tasks requiring context-sensitivity, it may make sense to rely on algorithms with recurrent dynamics and feedback.

Summary

- The brain has multiple biologically plausible options for approximate inference. These are not mutually exclusive.
- Different algorithms could be used by different parts of the brain, based on their complementary strengths and weaknesses for different tasks. For tasks requiring fast sensory processing, it may make sense to rely on primarily feedforward algorithms, whereas for tasks requiring context-sensitivity, it may make sense to rely on algorithms with recurrent dynamics and feedback.
- Another possibility is that these algorithms are integrated; for example, there are ways to use sampling methods in the service of variational inference and predictive coding.

Summary

- The brain has multiple biologically plausible options for approximate inference. These are not mutually exclusive.
- Different algorithms could be used by different parts of the brain, based on their complementary strengths and weaknesses for different tasks. For tasks requiring fast sensory processing, it may make sense to rely on primarily feedforward algorithms, whereas for tasks requiring context-sensitivity, it may make sense to rely on algorithms with recurrent dynamics and feedback.
- Another possibility is that these algorithms are integrated; for example, there are ways to use sampling methods in the service of variational inference and predictive coding.
- The fact that evidence exists for all of these possibilities suggests that the complete picture is likely complex, not reducible to any single simple algorithm.