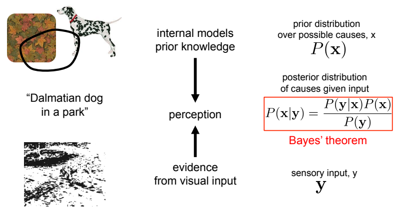
The army of one (sample): the characteristics of sampling-based, probabilistic neural representations

Pietro Berkes[1], **Richard E. Turner**[2], **József Fiser**[1,3]

[1] Volen Center for Complex Systems, Brandeis University [2] CBL, Dept Engineering, University of Cambridge, UK [3] Dept of Psychology and Neuroscience Program, Brandeis University

The Bayesian framework for perception

The brain makes use of internal models of the environment in order to resolve ambiguities and estimate uncertainty.

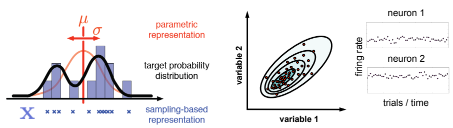


Behavioral evidence:

- Classical conditioning (Courville *et al.* TICS 2006)
- Perceptual processes (Kersten *et al.* Ann. Rev. Psych. 2006)
- Visuo-motor coordination (Kording & Wolpert, Nature 2004)
- Cue combination (Atkins *et al.* Vis Res 2001; Ernst & Banks Nature 2002)
- Decision making (Trommershäuser *et al.* TICS 2008)
- High-level cognitive processes (Griffiths & Tenenbaum, TICS 2006)
- Visual statistical learning (Orban et al. PNAS 2008)

Probabilistic computations in the brain

How do neurons represent and compute with probability distributions?



	Parametric	Sampling-based
Neurons represent	Parameters	Value of variables
Network dynamics	Deterministic	Stochastic
Representable distribution	Belongs to a parametric family	Arbitrary
Critical factor in encoding a distribution	Number of neurons	Time allowed for sampling
Representation of uncertainty	Complete at any time	Partial, requires sequence of samples
Number of neurons for multimodal distribution	Exponential	Linear
Learning	Updates are complex functions of parameters **e.g., PPC: Previously unknown See posters II-52 (Turner et al.), III-44 (Beck et al.)**	Well-suited e.g., Helmholtz machine

Sampling-cased representation remain unexplored, but are compatible with a number of experimental observations:

- trial-by-trial variability
- spontaneous activity (Berkes, Orban, Lengyel, Fiser, 2011)
- compatible with human behavior in single trials: "one and done" (Vul et al., 2009)

Sampling-based representations: open questions

Sampling has great asymptotic properties: unbiased, represents arbitrary correlations in multi-dimensional, multi-modal distributions. The brain needs to make decision in real time in a constantly fluctuating environment. Is this proposal for neural representation of uncertainty viable in practice?

Frequently asked, open questions:

1) **How can neural circuits generate samples from a particular internal model?**
2) **How many (independent) samples are required to make accurate estimates?** How long does it take to generate independent samples?
3) **What happens when the input is not stationary?** Is it possible to obtain accurate estimates then? How do the dynamics of the Markov chain operator interact with the temporal dynamics of the stimulus?
4) **Does a limited number of samples lead to a bias in learning?**

1) How can sampling be implemented in neural circuits?

Sampling equations result in plausible NN architectures and dynamics.

Gibbs sampling: The state of a neuron is sampled conditioned on the (estimate of) the state of the other neurons and the current input:

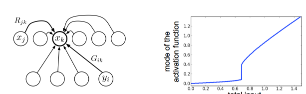
$$x_k \sim P(x_k|x_{\neg k}, \mathbf{y})$$

For example, for sparse coding model:

$$P(x_k) = P_{\text{sparse}}(x_k) \propto \exp(f(x_k))$$
$$P(\mathbf{y}|\mathbf{x}) = \text{Norm}(\mathbf{y}; \mathbf{G}\mathbf{x}, \sigma_y^2)$$

Gibbs sampling equations can be turned into simple neural network architecture:

$$\underbrace{p(x_k|x_{i\neq k}, \mathbf{y})}_{\text{Stochastic neural activation function}} \propto \exp\left(\frac{1}{\sigma_y^2}(\underbrace{\sum_i G_{ik}y_i}_{\text{Feed-forward weights}})x_k + \frac{1}{\sigma_y^2}(\underbrace{\sum_{j\neq k} R_{jk}x_j}_{\text{Recurrent weights}})x_k - \frac{1}{2\sigma_y^2}x_k^2 + f(x_k)\right) \quad \text{with} \quad \mathbf{R} = -\mathbf{G}^T\mathbf{G}$$



(Berkes et al., 2009)

Hamiltonian Monte Carlo: augment model variables with 'momentum variables', in analogy with physical system

Langevin sampling: special case of Hamiltonian MC; following dynamics for a single step at each iteration, one can get rid of the momentum variables, which results in this dynamical equation:

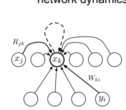
$$P(\mathbf{x}) = \frac{1}{Z}\exp(-E(\mathbf{x}))$$

$$\mathbf{x}(\tau+\epsilon) = \mathbf{x}(\tau) - \frac{\epsilon^2}{2}\frac{\partial E}{\partial \mathbf{x}}(\mathbf{x}(\tau)) + \epsilon\mathbf{n}(\tau)$$

defines a neural network dynamics

For example, for a Linear Dynamical System (Kalman filter):

$$P(\mathbf{x}_t|\mathbf{x}_{t-1}) = \text{Norm}(\mathbf{x}_t; \Lambda\mathbf{x}_{t-1}, \Sigma)$$
$$P(\mathbf{y}_t|\mathbf{x}_t) = \text{Norm}(\mathbf{y}_t; \mathbf{W}\mathbf{x}_t, \sigma_y^2)$$

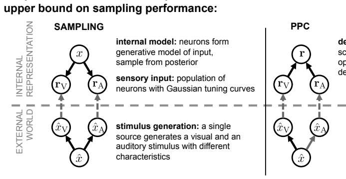


Corresponding Langevin dynamics:

$$\mathbf{x}_t(\tau+\epsilon) - \mathbf{x}_t(\tau) = \underbrace{\frac{\epsilon^2}{2\sigma_y^2}}_{\text{Feed-forward weights}}\mathbf{W}^T\mathbf{y}_t - \underbrace{\left(\frac{\epsilon^2}{2\sigma_y^2}\mathbf{W}\mathbf{W}^t + \Sigma^{-1}\right)}_{\text{Recurrent connections}}\mathbf{x}_t + \underbrace{\left(\frac{\epsilon^2}{2\sigma_y^2}\Sigma^{-1}\Lambda\right)}_{\text{Adaptation}}\mathbf{x}_{t-1} + \epsilon\mathbf{n}_t$$

Sampling-based cue combination model

Simple model, well-studied in parametric case with Probabilistic Population Codes, upper bound on sampling performance:



Ma et al., 2006; cf. **poster II-52**

Internal model: static / dynamic, Gaussian source

$$P(x) = \text{Norm}(x; \mu_{\text{prior}}, \sigma^2_{\text{prior}})$$
$$P(x_t|x_{t-1}) = \text{Norm}(x_t; \Lambda x_{t-1}, \Sigma)$$

Sensory neurons: Gaussian tuning curves $\text{M} \in \{\text{A, V}\}$

$$f_{\text{M},j}(x) = g_{\text{M}}\text{Norm}(x; \mu_{\text{M},j}, \sigma^2_{\text{M}})$$
$$P(r_{M,j}|x) = \text{Poisson}(r_{M,j}; f_{\text{M},j}(x))$$

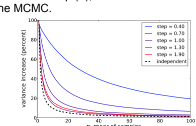
Neurons are sampling from $P(x_t|\mathbf{r}_{\text{V}}, \mathbf{r}_{\text{A}})$ using a Langevin dynamics:

$$\frac{\partial E}{\partial x}(x) = -\frac{(x-\mu_{\text{prior}})}{\sigma^2_{\text{prior}}} + \sum_i \frac{r_{\text{A},i}}{f_{\text{A},i}(x)}\frac{\mathrm{d}f_{\text{A},i}}{\mathrm{d}x} + \sum_j \frac{r_{\text{V},j}}{f_{\text{V},j}(x)}\frac{\mathrm{d}f_{\text{V},j}}{\mathrm{d}x} - \sum_i \frac{\mathrm{d}f_{\text{A},i}}{\mathrm{d}x} - \sum_j \frac{\mathrm{d}f_{\text{V},j}}{\mathrm{d}x}$$

2) How many samples for accurate estimate?

Estimation using samples is unbiased and asymptotically optimal. If extreme precision is not needed, a handful of samples can be enough.

How does the variability of an estimation computed with a small number of samples compare to the the optimal Maximum Likelihood estimator? The asymptotic behavior is 1/sqrt(T), but there is an additional scaling factor due to the dynamics of the MCMC.



With 1 sample, the variance of the estimate is just twice the optimal (ML) variance

epsilon	indep.	0.4	0.7	1.0	1.3	1.9
<25%	4 +/- 0	75	22	10	6	7
<10%	10.3 +/- 0.46	208	63	26	15	18
<5%	20.4 +/- 0.49	419	130	50	31	38
r_1	0	0.9	0.7	0.42	0.17	0.28

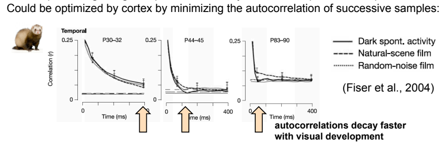
Independent sampler, 50'000 samples, 10 runs

Langevin sampler, 50'000 samples, 5 runs, after burn-in (300 iterations)

$$r_1 = \frac{\langle(x_t - \bar{x})(x_{t-1} - \bar{x})\rangle}{\langle(x_t - \bar{x})^2\rangle}$$

first-order autocorrelation

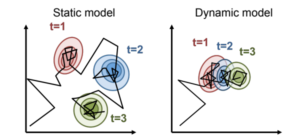
Best step for Langevin gives performance very close to independent sampler.
Could be optimized by cortex by minimizing the autocorrelation of successive samples:



(Fiser et al., 2004)

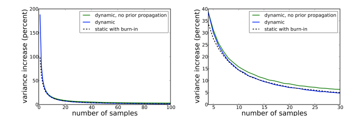
3) What happens for non-stationary input?

A time-varying input is not a problem if the internal model capture its dynamics. Benefits: no burn-in, tighter posterior, deal with missing data (e.g., occlusion).

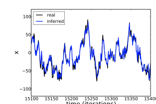


$$P(x_t|x_{t-1}, y_t) \propto \underbrace{P(y_t|x_t)}_{\text{same as in static case}}\underbrace{P(x_t|x_{t-1})}_{\text{tighter than } P(x)}$$

Results with dynamical model, ε=1.3, T=50'000, w/ and w/o propagating uncertainty information, no burn-in



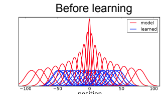
	static w/ burn-in	dynamic, no propagation	dynamic
<25%	6	7	6
<10%	15	17	15
<5%	31	40	30



4) Is learning possible with a small number of samples?

Accurate learning is possible even with a very small number of samples.

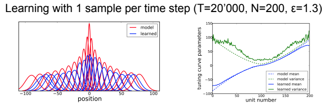
Before learning



Online learning in dynamical model, using EM:

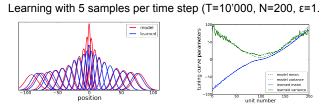
$$\arg\max_{\theta}\langle\log P(x, \mathbf{r}_{\text{V}}, \mathbf{r}_{\text{A}}|\theta)\rangle_{Q(x)}$$
$$= \arg\max_{\mu,\sigma^2}\sum_k \log P(\mathbf{r}_{\text{V}}|x^{(k)}, \mu_{\text{V}}, \sigma^2_{\text{V}}) + \log P(\mathbf{r}_{\text{A}}|x^{(k)}, \mu_{\text{A}}, \sigma^2_{\text{A}})$$

Learning with 1 sample per time step (T=20'000, N=200, ε=1.3)



Some boundary effects due to under-estimating uncertainty

Learning with 5 samples per time step (T=10'000, N=200, ε=1.3)



Conclusions

- Using a sampling-based model including temporal dependencies, we were able to reproduce previous results of parametric models on a cue-combination task.
- Sampling is a highly plausible candidate: its performance is comparable to an optimal ML estimator even for a small (~ 1-30) number of samples.
- Learning can be done efficiently with just a few samples, and the learning equations are a simple function of neural activity (cf. Poster II-52).
- The current results represent an upper limit for sampling performance in realistic models.