

Accurate and adaptive neural recognition in dynamical environments

Li Wenliang, Eszter Vertes, Maneesh Sahani

Summary Survival often depends on the ability to accurately track and learn the behaviour of a latent feature in the world—for example, the movement of a predator or prey—from noisy and incomplete sensory data. The statistical computations that would guarantee accuracy are well understood. As new noisy sensory information arrive, an agent or organism must integrate this information with its current belief about the latent quantity to form an updated representation. The form of this update depends on an internal model of how the quantity being tracked is likely to change in time. In general, this model will itself be uncertain, and thus must be adapted on the basis of the incoming sensory data (and the updated beliefs inferred therefrom) to ensure consistency with external world. In all but the most trivial of cases, accurate internal models and beliefs must be probabilistic, and cannot be summarised by the evolution of simple point estimates, such as the mean, alone. Here, we propose a biologically plausible inference and learning algorithm that relies on a deterministic representations of probabilistic beliefs using the distributed distributional code (DDC; Vertes and Sahani, 2018). We demonstrate empirically that the flexibility of the DDC makes possible accurate inference and learning in nonlinear settings; that synaptic weights for computation can be trained using a biologically plausible delta rule; that no explicit distributional assumptions are required; that the statistics of the latent variables are readily accessible from the representation; and finally that recognition can adapt to new problems through learning a flexible internal model of the observations.

Details The brain needs to process a stream of noisy and ambiguous signals in the dynamical environment, extracting unobserved but biologically relevant information in real time. The computations that underlie such an ability can be understood using the framework of optimal inference under a statistical generative model of the observations. We assume that the brain can simulate generative models expressed as a discrete-time exponential family state-space model with latent variable \mathbf{z}_t :

$$p(\mathbf{z}_t|\mathbf{z}_{t-1}) = \exp [g_z(\mathbf{z}_{t-1}, \boldsymbol{\theta}_z) \cdot T_z(\mathbf{z}_t) - \Phi(\mathbf{z}_{t-1}, \boldsymbol{\theta}_z)] \quad (1)$$

$$p(\mathbf{x}_t|\mathbf{z}_t) = \exp [g_x(\mathbf{z}_t, \boldsymbol{\theta}_x) \cdot T_x(\mathbf{x}_t) - \Phi(\mathbf{z}_t, \boldsymbol{\theta}_x)] \quad (2)$$

where g is a natural parameter that depends on the model parameters $\boldsymbol{\theta}$ and the conditioning variables, T is a vector of sufficient statistics and Φ is the log normaliser.

How can we do inference on this model? We illustrate our approach with a hypothetical example of a predator tracking a prey that moves between the two commonly visited locations \mathbf{z}_t . In order to accurately track its true location efficiently in real time given noisy sensory data \mathbf{x}_t , such as sound caused by moving in a forest, the predator needs to solve the filtering problem: given the belief about the the prey’s location based on previous observations, update the belief when new sound arrives. It is then necessary for the predator’s brain to represent the posterior distribution of location $p(\mathbf{z}_t|\mathbf{x}_{1:t})$ which can be accurately updated and allows optimal computations of downstream tasks, such as planning to catch the prey.

Distributed representation We propose that neurons in the predator’s brain represent in their firing rates the uncertainty about the prey’s locations $p(\mathbf{z})$ in a DDC [2], that is by nonlinear moments or expectations of a set of nonlinear functions: $\mathbf{r}_z = \int \boldsymbol{\psi}(\mathbf{z})dp(\mathbf{z}) = \langle \boldsymbol{\psi}(\mathbf{z}) \rangle$. Such a representation makes it possible to approximate the expectation of other functions in the downstream computation linearly: if $h(\mathbf{z}) \approx \sum_i \alpha_i \psi_i(\mathbf{z})$, then $\langle h(\mathbf{z}) \rangle_{p(\mathbf{z})} \approx \sum_i \alpha_i r_i = \boldsymbol{\alpha}^\top \mathbf{r}_z$.

DDC filter To compute a DDC of the prey’s location, assuming for now that the predator has a perfect internal model of the prey’s movement and sound generation, the predator would first simulate these signals in the brain, producing neural activities that correspond to location $\boldsymbol{\psi}(\mathbf{z}_t)$ and those that correspond to auditory inputs $\boldsymbol{\phi}(\mathbf{x}_t)$ (Figure 1A, Left). A recognition network is then trained to predict $\boldsymbol{\psi}(\mathbf{z}_t)$ from $\boldsymbol{\phi}(\mathbf{x}_t)$ and the previous belief of the prey’s location \mathbf{r}_{t-1} . When the squared error is minimised, this network predicts the posterior means of the latent activations given the simulated inputs and the previous belief.

These posterior means form a DDC for the posterior distribution of \mathbf{z}_t given sound history (Figure 1A, Middle).

The optimal recognition network can be approximated by a simple form: $r_t^{(i)} = \sum_{j,k} W_t^{(ijk)} r_{t-1}^{(j)} \phi_k(\mathbf{x}_t)$. This computation can be implemented by an intermediate neural population that combines activities encoding the previous belief and neural responses to sensory input, which are then linearly weighted by the filtering operator \mathbf{W} . The filtering operator is trained using the delta rule given examples provided by sampling and bootstrapping: at each time step, the predator generates triplets $\{\mathbf{z}_{t-1}, \mathbf{z}_t, \mathbf{x}_t\}$ from the generative model and also use the previously estimated DDC \mathbf{r}_{t-1} . On a nonlinear dynamical system adapted from [1], we find that the DDC filter estimates the posterior mean of the prey’s location nearly optimally over a wide range of parameters $\boldsymbol{\theta}$ (Figure 1B,C).

Adapting to new dynamics The predator is able to adapt to the dynamics generated by different kinds of prey, and the internal model is likely to have a flexible structure to simulate many movement patterns. We assume that this flexible dynamics has a mean function in the form of a linear readout from Gaussian-shaped tuning functions. The predator would then learn the dynamics of different types of prey by adjusting the readout weights. We simulate learning of the predator using an online sequence of observations from the nonlinear SSM in [1], assuming that the predator knows only the dimensionality of the latent variable, functional form of the sound generation mechanism but not its parameters $\boldsymbol{\theta}_x$ which also need to be learned. The posterior statistics required for learning $\boldsymbol{\theta}$ are again approximated from the DDC basis functions Figure 1A (Right). After each update in $\boldsymbol{\theta}$, the filtering operator \mathbf{W} is adjusted using samples given the new $\boldsymbol{\theta}$ and previous belief, resulting in a wake-sleep like algorithm. Figure 1D-G show the result of learning.

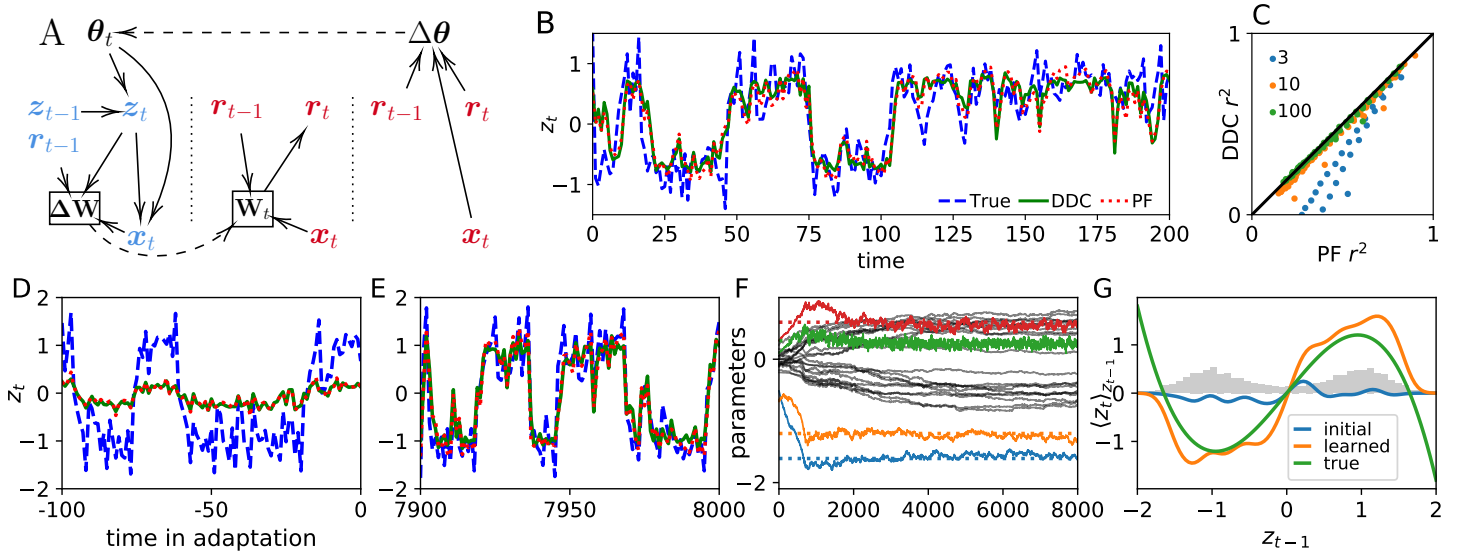


Figure 1: (A) (Left) Samples from the generative model and previous belief on simulated sequences (blue) are used to train the filter operator. (Middle) Filtering on the observed sequence given a real (red) observation (also performed on simulated observations to update belief). (Right) Adaptation using statistics calculated from posterior beliefs and observations. (B) Posterior mean of the DDC filter follows closely to that of a particle filter (PF). (C) Variance in the latent \mathbf{z}_t explained by PF and DDC filter with 3, 10, or 100 samples from generative model with different parameters. (D) Adapting the generative model parameters to a new prey. (Left) Before learning, the DDC filter and PF are equally bad at tracking \mathbf{z}_t . (E) After learning, the DDC filter accurately tracks \mathbf{z}_t and performs close to PF given the true generative parameters. (F) Parameters during learning. Colored: learned (solid) and true (dotted) values of $\boldsymbol{\theta}_x$. Black: linear weights on Gaussian tuning functions. (G) The learned mean function is close to the true one. Histogram shows distribution of \mathbf{z}_t

[1] A. Kutschireiter, S. C. Surace, H. Sprekeler, and J.-P. Pfister. *Scientific reports*, 2017.

[2] E. Vertes and M. Sahani. *COSYNE Abstracts*, 2018.