Nonlinear Bayesian Filtering in Artificial Intelligence

Rinat Galiautdinov

ORCID: 0000-0001-9557-5250

Abstract: A reliable assessment of dynamic latent characteristics, based on sensory inputs, is one of the signs of perception. This dynamic estimate can be formulated using the nonlinear Bayesian theory of filtration. Recent experimental and behavioral studies showed that animal performance in many tasks is consistent with this Bayesian statistical interpretation. However, it is currently unclear how a non-linear Bayesian filter can be effectively implemented in a network of neurons that satisfies some minimal limitations of biological certainty. Here the author offers the Neural Particle Filter (NPF), a non-linear Bayesian filter based on a sample that is independent of importance weights. The author shows that this filter can be interpreted as the dynamics of neurons in a recurrently connected neural network based on speed receiving a direct signal from sensory neurons. In addition, it covers the properties of temporal and multisensory integration, which are crucial for perception, and allows online study of parameters using the maximum likelihood method.

Keywords: Artificial Intelligence; Nonlinear Bayesian filtering

1. Introduction

After Helmholtz’s initial work, which hypothesized 150 years ago that this perception can be regarded as a process of unconscious inference, an increasing number of studies have shown that the brain performs perception tasks in accordance with Bayesian conclusion (Bain, A. & Crisan, D., 2009). From this point of view, perception relies on noisy and incomplete data, which must be integrated into various sensory modalities and weighed in accordance with sensory reliability. In addition, perception uses strict statistical laws of objects in our environment, forming preliminary ideas about the world. Since our environment is mostly dynamic, the ability to adapt to changes in real time is important for perception. This “Bayesian brain hypothesis” is confirmed by numerous experimental data: from psychophysical data to neural records which correspond to Bayesian calculations (Beck, J. M. & Pouget, A., 2007). Nevertheless, in most studies related to perception theory, rather simple problems are considered when observations are generated either from static hidden variables or from hidden variables with a discrete state space or the underlying dynamics are considered linear.

In a dynamic environment where time-varying signals must be evaluated online based on the observation history, the Bayesian conclusion is usually called “filtering”. In general, non-linear Bayesian filtering is a complex task even without the need for a plausible implementation of neural architecture. If the previous distribution is Gaussian, and observations with noise linearly depend on latent states, then the output problem is solved using the Kalman filter which has received considerable attention in the signal processing community and is becoming increasingly important in neurobiology, phenomenological modeling, for example, in sensorimotor integration problems or in the assessment of motor disorders by adaptive amplification. The solutions for most nonlinear, that is, non-Gaussian, filtration problems are analytically unsolvable and, therefore, should be approximated (Cappe, O., 2011).

At the algorithmic level, sampling-based approaches that represent sampling distributions have proven to be a powerful tool for numerically solving the nonlinear filtering problem. In principle, they make it possible to present any subsequent distribution with an accuracy that depends on the number of samples. On the one hand, the so-called particle filtering methods are well suited for dynamic a priori, but suffer from the inevitable weight loss over time, which leads to the “dimensional curse” (COD) in multidimensional models. A widely used strategy to reduce weight loss is re-sampling particles, but it can neither avoid COD nor increase its feasibility by a population of neurons. On the other hand, the Langevin sample and related methods, such as the “fast picker”, provide a promising basis for the biologically plausible implementation of neural or synaptic selection, but are limited to static generative models (Welling, M. & Teh, Y., 2011).

Following a sampling-based approach, the author offers a framework for how the brain can filter out noisy sensory stimuli. The author formulates perception as the task of assessing the dynamic state, which is posed in the context of the theory of nonlinear filtering with continuous time and continuous state. Based on this theory, the author proposes a particle filter without importance weights, a neural particle filter (NPF) (Yang, T. et al. 2016). The proposed filter can be implemented in a biologically realistic architecture using neural units based on speed. In particular, the absence of importance scales makes it possible to interpret this particle filter as neural dynamics: task-specific neurons are identified with samples from the back (or “particles”). Therefore, this method does not suffer from weight loss and related COD, which impede the biologically realistic implementation of suspended particle methods. To study the parameters, the author offers an online maximum likelihood method. This approach leads to nonlocal interactions in the learning rules for synaptic connections. However, the author finds that the rules for teaching hebbas are restored to the limit of low noise.

The author shows that the NPF algorithm demonstrates properties that are considered necessary for real-time
perception: taking into account both the noise of observations and sensory ambiguities, it weighs previous knowledge and sensory information from various modalities to form an estimate of the hidden state of the real world, and he is able to adapt his internal model in accordance with observations. As for the algorithmic estimation, the author numerically shows that, despite the fact that it is a technically non-optimal filter, NPF has an efficiency that is almost indistinguishable from optimal filters in low dimensions (Poyiadjis, G. et al., 2011). For large problems, the author gives numerical evidence that NPF does not suffer from COD and actually surpasses the methods of a weighted particle filter when the number of particles is limited. Thus, NPF can be considered as a candidate for a biological implementation of a filtering algorithm related to computing.

2. Background

Nonlinear filtering as a generic computational task

The author formulates the computational problem in terms of the classical filtering problem with additive noise. The latent state \( x_t \in \mathbb{R}^n \), which the brain cannot directly access, follows the Ito stochastic differential equation (SDE) (1):

\[
\frac{dx_t}{dt} = f(x_t) + \sum_{i} \frac{1}{2} \sigma_i \frac{dW_t}{dt}
\]

where \( f(x) \) is a vector-valued function that describes the drift of the state, \( \sigma_i \) are \( n \times n \) matrices that define the diffusion, \( \{W_t\}_{t \geq 0} \) is an \( n \)-dimensional Brownian motion process, \( t \in [0, T] \), which the brain cannot directly access, and \( \mathcal{C} \) is a non-random measurable set of \( \mathbb{R}^n \). The function \( f(x) \) is chosen so that there exists a constant probability density of the hidden state \( p(x_t) \), which serves as an a priori over the hidden state.

Stochastic diffusion is determined by the uncorrelated vector Brownian motion process \( w_t \in \mathbb{R}^n \) with noise covariance \( \Sigma \in \mathbb{R}^{n \times n} \).

At each moment of time, the latent state \( x_t \) causes noisy observations \( y_t \in \mathbb{R}^m \), which represent sensory input. The dynamics of the observations is again modeled in terms of Ito diffusion, with the drift term following the latent states through the generating function \( g(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m \) and the Brownian motion process \( u_t \), modulated by the covariance of sensory noise \( \Sigma_t \in \mathbb{R}^{m \times m} \) (2):

\[
\frac{dy_t}{dt} = g(x_t) + \sum_{j} \frac{1}{2} \sigma_j \frac{du_t}{dt}
\]

The functions \( f \) and \( g \) satisfy the standard conditions often used in nonlinear filtering.

Together, formulas (1) and (2) determine the generative model (Figure 1a). In particular, equation (1) leads to the probability of the transition \( p(x_t | x_{t-1}) \) of the latent state, and the formula (2) creates the probability of radiation (or the probability of observation) \( p(dy_t | x_t) \), respectively. Further, the author assumes that the function \( f(x) \) is chosen so that there exists a constant probability density of the hidden state \( p(x_t) \), as an a priori over the hidden state.

![Figure 1](image)

Generative model and neural network implementation. (a) The generative model defined by equations (1) and (2). (b) Execution of the formula (4) as a recurrent neural network. Left: each particle in NPF corresponds to one of \( N \) subnets that work in parallel. Here, each circle denotes a neural population. Right: connections between neurons in the \( k \)-th subnet in a model with a linear production function \( g(x) = Jx \). New neurons \( n_i \) are associated with filtering neurons \( z_i \) through the decoding weight \( W_{ij} \). Feedback connections from filter neurons \( z_i \) to new neurons \( n_j \) have a weight \( J_{ij} \). Further, the author assumes that the self-interaction is local, that is, the components of the hidden dynamics are independent: \( f(z^{(k)}) = f(z^{(k)}) \). Here, each circle represents one neuron (Galiautdinov, R. & Mkrtchian, V., 2019A).

The solution to the filtration problem is the problem of finding the a posteriori probability density of the hidden state \( p(x_t | Y_t) \), due to the entire sequence of observations \( Y_t = \{y_s, s \in [0, t]\} \) until time \( t \). For the linear hidden drift function \( f(x) \) and the linear observation function \( g(x) \), this problem is solved using the Kalman-Bucy filter, which is a version of the well-known Kalman filter with continuous time. However, the solution to the nonlinear filtering problem as a whole is analytically difficult to solve, since it...
suffers from the so-called closure problem. Therefore, the introduction of a suitable approximation is an inevitable step when approaching the problem of nonlinear filtering.

**View Based Sampling**

In the theoretical literature of neurobiology, sampling-based approaches for filtering are presented with a posteriori representation, as in the formula (3) so far not much attention has been paid, although they have some experimental support and are considered relevant in accordance with the “neural sampling hypothesis”. Therefore, the author would like to explore this approach further and, therefore, approximate probability density functions in terms of a finite number of variables. For example, this can be achieved by taking N weighted samples (3):

$$ p(x, t) = \sum_{k=1}^{N} w_t^{(k)} \delta(x - z_t^{(k)}) $$

With

$$ \sum_{k=1}^{N} w_t^{(k)} = 1 $$

Thus, the probability that a random variable will have a certain range of values is proportional to the relative number of samples in this range, weighted by their respective weight $w_t^{(k)}$.

Filtering algorithms representing a posteriori in this sample-based method are commonly referred to as particle filters (PF). Standard PFs contain update rules for sample (or “particles”) trajectories $z_t^{(k)}$, as well as weights $w_t^{(k)}$. Despite the asymptotic convergence to the true posterior for an infinite number of particles, this approach has two drawbacks: first, it is found numerically that after a finite number of time steps, the mass of most particles drops to zero, which reduces the number of effective samples (Galiautdinov, R. 2020). Weight reduction is an undesirable feature of particulate matter methods in general. Secondly, the problem is aggravated if the number of measurements of the latent state $x_t$ is large. In this case, the number of particles required for good numerical performance grows exponentially with the number of measurements being a variant of COD. The first problem is usually solved by re-sampling, and the second problem was solved, but the question remains how these purely numerical strategies can be implemented by neural units. To get around this issue, the author considers a particle filter with equally weighted samples, i.e. $w_t^{(k)} = 1/N$ or all $k$.

**Filtering with a filter of neural particles**

As an output algorithm, the author proposes SDE, which controls the particle dynamics $z_t^{(k)}$. Let’s look at $N$ i.i.d. random processes $z_t^{(k)}$, $k = 1, ..., N$, due to observations of $Y_t$ after Ito diffusion (4):

$$ dz_t^{(k)} = f(z_t^{(k)})dt + W_t(dz_t^{(k)} - g(z_t^{(k)})dt) + \sum_{x} \frac{1}{2} dw_x $$

where $w_t \in \mathbb{R}^n$ is the uncorrelated vector process of Brownian motion, and $W_t \in \mathbb{R}^{n \times m}$ is the time-dependent gain matrix or the matrix of decoding weights.

Formula (4), which the author will call the filter of neural particles (NPF), is an ansatz that serves as an approximation based on a sample to the nonlinear filtering problem: each of the $N$ random processes $z_t^{(k)}$ is conditionally independent empirical sample or particle from the distribution

$$ p(x_t) = 1/N \sum_{k=1}^{N} \delta(x_t - z_t^{(k)}) $$

, which approximates the true a posteriori $p(x_t \mid Y_t)$ at each time $t$, i.e.,

$$ p(x_t \mid Y_t) \approx 1/N \sum_{k=1}^{N} \delta(x_t - z_t^{(k)}) $$

Expectations regarding the rear (*) are approximated in accordance with (5):

$$ E[\phi(x_t)Y_t] = (\phi(x_t)) \approx \frac{1}{N} \sum_{k=1}^{N} \phi(z_t^{(k)}) $$

Ansatz in formula (4) is motivated by a formal solution to the filtration problem, more precisely, by the dynamics of the first posterior moment, and has some important properties with classical filtration methods: firstly, it is determined by both the dynamics of the hidden process $x_t$ and using correction proportional to the so-called innovative term $dn_t^{(k)} = dy_t - g(z_t^{(k)})dt$. The innovative term compares the touch input $d_y$, with the current forecast $g(z_t^{(k)})dt$ in accordance with the position of an individual particle and, thus, can be considered as a signal of a predictive error. Secondly, the gain matrix $W_t$ determines the emphasis placed on new information through observations of $d_y$. Conceptually, this is similar to Kalman’s gain for a linear model.

The gain $W_t$ can, for example, be calculated in accordance with

$$ W_t = \text{cov}(x_t, g(x_t))^{-1} $$

by empirical choice, motivated by formula S-16. This gain is adjusted according to the observation noise $\Sigma_y$, as well as the spatial ambiguity measured by the covariance between the state $x_t$ and the generating function $g(x_t)$. This covariance cannot be obtained directly, but is estimated empirically and instantly from the positions of the particles through the formula (5). The resulting filtering algorithm with an empirically determined gain is summed in algorithm (1).

The gain factor introduces weighting between the a priori probability distribution $p(x_t)$ induced by the formula (1) and the likelihood function $p(d_y \mid x_t)$ induced by the formula (2)
and, thus, serves as a measure for the peak of probability. If the observation noise is small, the decoding weight is large. Then the dynamics in the formula (4) will be completely determined by the innovative term, and the variability between particles, determined by the diffusion term, will be negligible. In this limit, the deterministic limit of observation, one sample from the formula (4) enough to represent the back. On the other hand, if the weight of the decoding is zero, the new information is ignored, and each sample develops exactly like i.i.d. copy of equation (1). In this case, the resulting a posteriori density is simply equal to the stationary a priori density \( p(x_t) \).

**Algorithm 1**

An algorithm for filtering neural particles with an empirical gain function. The URF continuously extracts hidden objects \( z_t \) from the input stream of observations \( y_t \). The distribution of characters is represented by N populations of neurons.

1. NPF procedure (\( \{z_t, δ_t^{(k)}\}_{k=1}^{N}, \delta y_t \))
2. Computes a weight matrix that connects new neurons to filter neurons (6):
   \[
   W_t = \frac{1}{N} \sum_{k=1}^{N} z_{t-δ_t}^{(k)} g(z_t-δ_t) - \frac{1}{N^2} \sum_{k,l=1}^{N} z_{t-δ_t}^{(k)} g(z_{t-δ_t}^{(l)})
   \]
3. for \( k = 1 \) to \( N \) do
4. update novelty neurons \( n^{(k)} \) (7):
   \[
   \delta n^{(k)}_t = \delta y_t g(z_t - \theta) dt
   \]
5. update filter neurons \( z^{(k)}_t \) (8):
   \[
   z^{(k)}_t = z_{t-δ_t}^{(k)} + f(z_{t-δ_t}^{(k)}) δt + W_t δn^{(k)}_t + \sum_{k'} δW^{(k)}_t
   \]
   With
   \[
   δW^{(k)}_t \sim N(0, δt)
   \]
6: end for
7: return \( \{z^{(k)}_t\}_{k=1}^{N} \)

8: end procedure

**Training parameter**

In the more general case, the model parameters \( \theta \) of formulas (1) and (2) may not be known or only partially known, and therefore, they must be studied online from the observation flow \( Y_t \). In this case, the NPF algorithm can be expanded to include an update to the parameter, which performs an online gradient ascent along the logarithmic probability of the entire observation history. \( Y_t \) (9):

\[
L_t(\theta) = \int_0^T \delta y_t g(x_t) dx_t > g T \sum_{y} \delta y_t g(y) dx_t > g^T \sum_{y} \delta y_t g(y) dx_t > g ds
\]

which, in turn, is calculated directly from the approximate distribution of filtration through empirical estimates (formula 5). Note the dependence of the conditional estimate on the parameters of the \( \theta \) model, which must be taken into account when trying to maximize the logarithmic probability. It can be shown that maximizing this logarithmic probability is equivalent to minimizing the prediction error in continuous time.

The parameters of the \( \theta \) model are studied by online gradient rise in logging probability, which leads to the following training rules for the parameters \( \theta \) (10):

\[
\eta_\theta \frac{\partial}{\partial \theta} s_t \sum_{y} \delta y_t g(y) dx_t > g T \sum_{y} \delta y_t g(y) dx_t > g^T \sum_{y} \delta y_t g(y) dx_t > g ds
\]

This approximation of online learning is justified if the learning timeline is much larger than the filter dynamics, that is, for low learning speeds.

In formula (10), the novelty signal \( \delta y_t - g(x_t) dt \) is multiplied by the parameter gradient on the posterior estimate of the generating function \( g(x_t) \). Thus, the author must take into account the implicit change in the posterior distribution with respect to the model parameters, the so-called filter derivative. Filter controllability problems also apply to filter derivatives, and the latter should be approximated by particle representations.

Learning rules for the parameters in the formula (10) are nonlocal, i.e. they depend on the aggregated states of the entire set of particles. In the limit of low observation noise and for the linear observation model \( g(x) = Jx \), the learning rule for the generating matrix \( J \) (or the mixing matrix) can be approximated by the local and Hebbsky training rule (11):

\[
dφ(\delta y_t - f(x_t) dt) (x_t) \approx (\delta y_t - f(x_t) x^2)
\]

It is important to note that not only the model parameters in formulas (1) and (2), but also the decoding parameters, that is, the components of the gain matrix \( W_t \), can be studied with the maximum likelihood approach, in contrast to the setting obtained in accordance with an empirical estimate from the position of the particles. The training rules for the components of the gain matrix read (12):

\[
dW_{ij}^{r} \eta_r \frac{\partial}{\partial W_{ij}} s_t \sum_{y} \delta y_t g(y) dx_t > g T \sum_{y} \delta y_t g(y) dx_t > g^T \sum_{y} \delta y_t g(y) dx_t > g ds
\]

This alternative to determining gain corrects the heuristic ansatz of the NPF equation by rigorously decoding the decoding weights. In fact, it can be shown that studying parameters with a maximum likelihood approach can compensate for even a very poor filtering ansatz by adjusting the parameters accordingly.

**The implementation of neurons**

The NPF can be interpreted in terms of the dynamics of neurons, and the algorithm can be implemented in a recurrent neural network (Huang, Y. & Rao, R., 2016). In particular, the author considers the dynamics of a population of \( N \times n \)-filtering neurons \( z^{(k)}_t \), whose analog neural activities (e.g., instantaneous firing speed) represent samples of the back, in accordance with the hypothesis of neuronal sampling (Figure 1b). The calculation is performed on N subnets, one for each "particle" in the NPF. The architecture of each parallel subnet is structurally similar to the architecture presented in Rao & Ballard. As input data for subnets, the author considers the neural population \( y_t \), the
In each subnet k, there are two types of neurons, novelty neurons $n^{(k)}$ and filtering neurons $z^{(k)}$. A population of new neurons $n(k)$ receives information from a population of sensory neurons $y$ and is recurrently associated with a population of filtering neurons $z^{(k)}$ (see formula (7)). The output signal of the new neuron $n^{(k)}$ is the difference between the actual sensor input and the expected input in this subnet (single-particle prediction error) (Kording, K. et al, 2007). The output of novelty neurons is taken by filtering neurons $z^{(k)}$ through synaptic weights with a direct connection $W_i$ (see formula 8). Consequently, the dynamics of filter neurons depends both on sensory inputs (through novelty neurons) and on previous dynamics (through the nonlinear function $f(z^{(k)})$). The output of filtering neurons in each subnet corresponds to the state of an individual particle in the particle filter (Galiautdinov, R. 2019).

In this implementation, $W_i$ corresponds to a synaptic balance matrix that connects novelty neurons $n^{(k)}$ with filtering neurons $z^{(k)}$. If the derivative function $g(x_i) = Jx_i$ is linear, then $J$ denotes the feedback weight matrix that connects filtering neurons to novice neurons (Figure 1b, right). In general, the training rules for these weights that arise from maximizing probability in a formula (9) are not local, that is, they depend on the state of the entire network (formula 10). However, in the deterministic limit, the learning rule for the matrix of generative weight $J$ can be replaced by the learning rule, which is both Hebb and local, based on the multiplication between pre- and postsynaptic activity (formula 11), i.e. between filtration and novelty neurons. In addition, with a small observational noise, $W_i$ can be replaced by a constant matrix without affecting the filtration performance (provided that the weights are large compared to the previous dynamics). Therefore, at least in this limit, the network shown in Figure 1b, it is realizable as the neural dynamics of a recurrent network with Hebb local synaptic plasticity (Peccevski, D., 2011).

### 3. Results

Following a top-down approach, the author examines how the brain can realize dynamic perception. First, the author interprets perception as a computational problem of nonlinear Bayesian filtering. The solution of the nonlinear filtering problem in the general case is infinite-dimensional and, therefore, requires a finite-dimensional approximation. Choosing a representation of the time-varying posterior in terms of empirical samples that are distributed in accordance with the NPF equation (4), the author can formulate such a finite-dimensional approximation. In addition, samples (or “particles”) are directly identified with the activity of filtering neurons and the NPF equation with their neuronal dynamics. Thus, the author can base the implementation of the algorithm on neural architecture.

Using a simple example, the author is going to illustrate that our algorithm captures the following key perceptual properties: (1) it relies on noisy and incomplete sensory data, (2) it uses preliminary knowledge about the dynamic structure of the medium, (3) it effectively combines information from several sensory modalities and (4) can dynamically adapt to changes in the environment.

**Perception as Nonlinear Filtering**

Consider an owl that sits under two branches and tracks a mouse moving between two branches (Figure 2a). The owl cannot directly observe the position $x_t$ of the mouse, which will be called the “latent state” in the future, but instead it has to rely on two sensory channels: the visual ($v_t$) and the auditory ($a_t$) channels. These observations are called a latent state through generating functions (Figure 2b), which enter the dynamics of observations in a deterministic drift (13):

\[
dv_t = x_t dt + \sigma v dt + \gamma w dt + \sigma w dt + \sigma a dt
\]

\[
da_t = \text{tanh}(2x_t) dt + \gamma a dt + \sigma a dt
\]

$\beta$ and $\gamma$ are independent Brownian motion processes that simulate sensory noise, making $v_t$ and $a_t$ conditionally independent. Nonlinearity in the auditory channel (formula 14) is motivated by the fact that the localization of sound depends on interpersonal differences, which the author models as sigmoid in this one-dimensional example.
A toy model illustrating the filtering task for perception. (a) a scene of a owl watching and catching a mouse, drawing on its visual $(v_t)$ and auditory $(a_t)$ channels. (b) Non-linearity in the generating function $g(x)$ of two sensory channels. Vision is modeled as a linear display, and listening is modeled as a sigmoid function (c). A sample of the flight path of a fly according to the formula (15). Note that the nonlinearity in the drift leads to a bimodal stationary distribution $p(x_t)$.

In addition, the owl has some prior knowledge about the dynamics of the fly, which the author models as (15):

$$dx_t = 3x_t(1-x_t^2)dt + dw_t;$$

where the Brownian motion process explains the noise due to the unstable behavior of the mouse. An example of the trajectory from this random process is shown in Fig. 2c. The author notes that the nonlinearity of the drift function in this dynamics leads to a bimodal stationary distribution of the position of the mouse.

To track a fly, the owl must combine information from its sensory input and combine it with previous knowledge in order to calculate the posterior density $p(x_t | V_t, A_t)$, i.e. the probability of finding a fly in a certain spatial region, taking into account visual and auditory sensory fluxes $V_t = \{v_s; 0 \leq s \leq t\}$ and $A_t = \{a_s; 0 \leq s \leq t\}$. This task is usually called nonlinear filtering. Due to the nonlinear dynamics of the processes of covert observation and observation, the solution of this particular example is analytically insoluble and, therefore, requires approximation (Churchland, A. K. et al., 2011).

The author proposes to solve this problem using a set of $N$ filtering neurons $z_t^{(k)}$, $k = 1, ..., N$. The empirical distribution of neural activities $z_t^{(k)}$ approximately measures the posterior density, thereby acting as a filter of particles without weight, which successfully tracks the position of the mouse (Figure 3a). An estimate of the state $x_t$ (a posterior mean) can be read from this population by averaging the activities of filtering neurons, i.e. $x_t^{(\text{av})} = N^{-1}\sum z_t^{(k)}$.

The NPF as a model of perception for multisensory perception. (a) Tracking simulation with $N = 1000$ filtering neurons and sensory noise $\sigma^2_v = \sigma^2_a = 0.1$. $\sigma^2_v = \sigma^2_a = 0.1$. The top panel shows the true trajectory of the mouse (solid black line) and particle density. The areas between the dashed black lines indicate two branches, and the confidence levels on the middle panel correspond to the relative number of particles whose states are in one of two branches. Each time, the sensory amplifications in the bottom panel are calculated in accordance with formulas (17) and (18). (b)
The neuronal dynamics of these filtering neurons are given by the NPF (formula 4) and for this particular example read (16):

\[
\frac{dz^{(k)}}{dt} = 3z^{(k)}(1 - (z^{(k)})^2)dt + dw^{(k)} + W^{(v)}(dv - z^{(k)}dt) + W^{(a)}a_t(dat - \tanh(2z^{(k)})dt)
\]

Firstly, this dynamics is determined by the dynamics of the fly (formula 15), which serves as a forecast in accordance with the fly's previous knowledge about the position of the fly. Secondly, the forecast is corrected by the novelty of observations in sensory channels. The novelty effect is modulated by the two components of the gain matrix \(W^{(v)}\) and \(W^{(a)}\). A possible network implementation of this dynamics is shown in Figure 1b.

With an approximate solution to the filtering problem, our model easily captures the first two key properties of perception, that is, it relies on noisy and ambiguous observation data and is modified by a preliminary knowledge of the dynamics of the fly (Galiautdinov, R. et al., 2019B).

The potential for describing a full posterior extends far beyond a simple state assessment when only the first moment is of interest. In particular, the approximation of this posterior based on the sample makes it possible to conveniently estimate other corresponding quantities. For example, a owl might want to know which branch an mouse sits on to make it easier to catch. He can directly derive the confidence level for the left and right branches, respectively (Figure 3a), by counting the number of samples within a certain range of activity.

**Cue integration**

Decoding weights or gains \(W^{(v)}\) and \(W^{(a)}\) are important for multi-sensor integration. They balance the relative influence of two sensory modalities and the previous level on the dynamics of filtering neurons and, thus, quantitatively determine the reliability of sensory channels. Here the author considers the empirically estimated gain in NPF. In this example, weights are estimated as (17), (18):

\[
W^{(v)} = \text{cov}(x_v, g(x_v)) \sum_y \frac{1}{\sqrt{\text{var}(x_v)\sigma^2_v}} \quad W^{(a)} = \text{cov}(x_t, \tanh(2x_v))\frac{1}{\sqrt{\sigma^2_a}}
\]

The gain is adjusted in accordance with the sensory noise levels, that is, the amplifications are reduced on average to increase the noise level within one channel (Figure 3b). In addition, they are corrected in accordance with the “spatial” uncertainty caused by the nonlinearity in the observation function (Mkrtchian V., Gamidullaeva L. & Galiautdinov R., 2019). More precisely, the gains are determined by the covariance between the state and the generating function, which is associated with the slope of the nonlinearity. Roughly speaking, the more the productive function \(g(x_t)\) changes relative to the (currently estimated) position \(x_t\), the more reliable is the observation. In this example, the sigmoid observation function of the auditory canal \(a_t\) is more reliable if the fly is considered to be in the center, and thus the auditory coefficient is adjusted accordingly (bottom panel in Figure 3a).

Thus, the gain becomes large if the channel is especially reliable, and in extreme cases, the dynamics of filtering neurons, corresponding to the deterministic limit of observation, dominate. Appropriate weighing of sensory information allows neurons to solve the filtering problem almost optimally and is comparable to the standard PF, which, for example, is reflected in our simulation results in Figure 3c.

**Adaptation of the internal model**

In our example, the owl could successfully track the position of the mouse, but it could only do this because it had access to the parameters of the generative model in its internal model, i.e. she knew the previous dynamics of the mouse and knew how sensory perceptions were created from the true state of the mouse. In addition, knowledge of these model parameters was crucial for determining the sensory weights \(W^{(v)}\) and \(W^{(a)}\). And, thus, significantly influenced the dynamics of filtering neurons. Nevertheless, the external world, represented by the parameters of the model, changes with time and successful perception, must adapt the internal model accordingly (Sokoloski, S., 2015).

The author illustrates the study of the parameters of the generative model in our example with maximum probability (ML, formula 10). In addition, the author considers the limit of small sensory noise, which leads to the training of Hevby in the neural network (Figure 1b, cf. formula 11). This time, the owl relies only on its visual channel \(v\), but in addition to tracking the mouse, it also needs to study the generation factor \(J\) in the function \(g(x) = Jx\), which links the position of the mouse for visual input. At the same time, he also recognizes the gain \(W^{(v)}\) in accordance with the formula (12) and with this, an implicit estimation of the reliability of its visual input. Figure 4 shows that this identification problem can be effectively resolved with the help of the URF, while the MSE is gradually approaching the reference one (standard PF with true parameters relative to the ground), since the parameter estimation becomes more accurate (Greaves-Tunnell, A., 2015).
The parameters of the model are studied by stochastic gradient rise according to the probability of the log. The simulation shown here corresponds to an exemplary model with only a visual hint, that is, a generative model with the formulas (13) and (15). The generative weight J is studied online by maximum likelihood (ML, formula 10) or the Hebb learning rule in the formula (11), which is a valid approximation for small sensory noise (Huang, Y. & Rao, R., 2014). The sensory amplification W is studied online simultaneously using the formula (12). As a guide, the author uses a weighted PF with true model parameters. For both parameters W and J, training starts at t = 0. When J’ approaches the true value J = 1, the trajectory of the filtering neurons (purple) can follow the trajectory of the true hidden state (black) and the MSE NPF resembles the MSE standard PF.

**Algorithmic Evaluation**

The NPF equation (formula 4) is structurally very similar to a filtering algorithm called an inverse particle filter. The main difference between the NPF described in Algorithm (1) and the FBPF with the constant gain approximation (used to enable the calculation of gain in higher dimensions) lies in another observation prediction used in the innovative term, so that NPF in accordance with the network implementation, as shown in Figure 1b. This important difference, however, does not adversely affect NPF performance compared to FBPF characteristics, as shown in Figure 5 for a linear model.
ILP avoids the “curse of dimensionality”. (a) Filtering performance in terms of MSE (normalized to optimal performance, in this case MSE^{opt} = 0.5d, where d denotes the number of hidden measurements) for a different number of particles for a linear model with a multidimensional hidden state - space (d = 80). Both unweighted NPF and FBPF approaches outperform the (standard) weighted particle filter for a limited number of particles (b). Particle number required to achieve numerical performance MSE <1.5MSE_{opt}. The number of particles N required for a standard (weighted) FS increases exponentially with increasing d due to the rapid attenuation of the mass in higher dimensions. Conversely, unweighted approaches avoid COD, and the number of particles scales linearly with hidden sizes. The solid lines correspond to the linear (N = a+d + b, NPF: a = 0.38, b = 4.1 FBPF: a = 0.45, b = 3.2) and exponential (N(d) = c_{0}d^{b} + c_{d}d + c_{0}; PF: c_{0} = 47, c_{1} = 0.07, c_{2} = -2.4, c_{3} = -42) minimum squares fit. (c, d) Same as (a, b), but for nonlinear latent dynamics with a bimodal stationary distribution. The smallest squares correspond to the coefficients: a = 0.42, b = 4.2 (NPF), a = 0.45, b = 2.8 (FBPF), c_{0} = 44, c_{1} = 0.07, c_{2} = -2.1, c_{3} = -3.8 (PF).

For the approach of weightless particle filtering, such as NPF (or FBPF), the author can say with confidence that scaling the required number of particles with a size smaller than exponential (Figure 5) and, thus, avoids COD in our modeling examples (Wilson, R. C. & Finkel, L., 2009). The authors find that NPF using works well even for a limited number of samples, and the number of particles needed for a certain productivity grows linearly with increasing latent sizes. The standard PF, however, exhibits exponential scaling of the number of particles, illustrating COD (Fiser, A. et al., 2016).

\[ W_t = \cos \left( x_t, g(x_t) \right) \sum_{\gamma}^{1} \]

In other words, despite the fact that NPF is a non-optimal filtering algorithm, a limited number of particles is sufficient to solve the filtering problem in higher dimensions with acceptable performance. This reliability for a smaller number of particles is mainly due to the direct influence of the observations d y t on the trajectory of the samples. In the unweighted approaches that the author shows here (NPF and FBPF), each state of the particle can be considered as a point estimate of the state, which becomes accurate for very small observation noise \Sigma_{x}. Of course, the more \Sigma_{x} becomes, the less the true posterior resembles the \delta-function, and the more particles are required to take into account its size as a whole.

Besides being just an algorithmic feature, scaling with measurements has biological significance. Consider, for example, our example from the previous subsection: until now, the latent state (position of the mouse) was considered purely one-dimensional. In a more realistic environment, the brain is confronted with a much larger number of latent states that it must bring out, starting from the position of the object in three-dimensional space and ending with the relative presence of the elements that make up the visual scene (Bergemann, K. & Reich, S., 2012). Therefore, any filtering algorithm used by a neural population for perception must be economical in its resources: an algorithm that needs an exponential number of filtering neurons with increasing dimension, that is, an algorithm that suffers from COD, will be destructive. Instead, the number of neurons needed to solve the filtering problem to a certain level of performance should scale well with the number of hidden variables, a requirement that is met using a weightless particle filtering approach such as NPF.

4. Discussion

In this article, the author establishes perception in the context of the computational problem of nonlinear Bayesian filtering. Based on the theory of non-linear filtering, the author proposed analog dynamics for particles (or neurons), which serves as a weightless particle filter, NPF. NPF inherently reflects important properties that are considered critical to perception, and can later be implemented as neural dynamics in a recurrent neural network. Thus, this can serve as a step towards understanding how perception can be realized in the brain at a conceptual level (Orban, G. et al., 2016).

The NPF equation the author proposes in the formula (4). It is especially suitable for phenomenological modeling of perception, because it shares some important properties with perception. Firstly, perception relies on noisy, often ambiguous and incomplete sensory data, such as that found in visual scenes, and uses them to comprehend the world, which is reflected in our model by outputting a hidden state variable (Berkes, P., 2011). Secondly, the brain must effectively combine various sensory signals to reduce ambiguity or ambiguity. In addition, he uses strict statistical patterns of the environment, taking into account previous knowledge. In NPFs, multisensory integration is implemented as a weighted sum of sensory input, where the weights of modalities balance their significance. In particular, the weight of one modality is adjusted in accordance with its reliability, that is, decreases with increasing ambiguity or the level of sensory noise. This has also been reported experimentally6. Since the preceding dynamics is directly included in the NPF, preliminary knowledge of the environment is automatically included and, in principle, can be studied. Finally, perception must be able to adapt to changes in the environment, which is taken into account with dynamic amplification and online parameter updates (Rebeschini, P. & Van Handel, R., 2015).

There are two main competing proposals on how probability distributions underlying Bayesian calculations can be represented in the brain. First, it has been suggested that probability distributions are expressed as probability 40 population codes (PPCs), in which each neuron represents a state of a coded random variable, and their activity is proportional to the (logarithmic) probability of the corresponding state. Filtering approaches based on population codes have been studied in the literature for a large set of models. In this view, neurons directly correspond to the distribution parameters, and therefore, the number of neurons is a critical factor for accuracy. In addition, they all suffer from COD for multimodal distributions (Makin, J. G.et al., 2015).
The second sentence, called the neural sampling hypothesis, uses a logical inference scheme in which the activity of each neuron is a sample of the base probability density. This choice of representation is an important aspect applied by the NPF. At the neuron level, there was some support for the fact that neurons could indeed represent stimulus uncertainty in terms of samples. For example, it was shown that the variability of neurons between trials decreases after the onset of the stimulus, and that this may be associated with a decrease in perception uncertainty (Surace, S. C. et al., 2017). Moreover, it was found that spontaneous neural activities are related to previous expectations about the stimulus in the visual cortex. Since our filtering algorithm is based on unweighted samples, our conclusions are consistent with the benefits of the sample-based presentation presented by Fiser et al: it can represent any distribution without the need for a parametric form, it avoids COD and is well suited for learning. Filtering approaches that implement Monte Carlo algorithms with a Markov chain (MCMC) have recently received some attention, but since they are based on a discrete state space and suggest a coding scheme different from that proposed in Fiser et al., the benefits listed here do not necessarily arise from these models (Surace, S. C. & Pfister, J.-P., 2016).

As a filtering algorithm, NPF is comparable to existing sampling-based filtering approaches. Our ansatz can be considered as a particle filter in which all particles have the same weight and therefore avoid pitfalls such as weight degeneration. This problem is well known in standard MCMC particle filters and becomes even more serious with the increasing number of covert measurements. COD, i.e., an exponential increase in the approximation error with the dimension of the base model, is an inevitable obstacle in standard MCMC approaches. Weight degeneration can be slowed down by re-sampling the particles or using a finer particle propagator. However, none of the solutions can reduce weight loss in general. There are more sophisticated approaches to particle filtering, which are based on spatial localization and exhibit subexponential scaling in particles with increasing sizes. Currently, no implementation of such a filtering strategy or even for standard methods of suspended particles in neural architecture is proposed. Since the NPF is not primarily dependent on weights of importance, it does not suffer from these quantitative errors and related implementation problems. It seems that CODs are avoided due to the fact that the observations directly enter the particle trajectories, and do not lead to an even faster decrease in weight, but, of course, the numerical estimate presented here does not provide general evidence that NPF avoids COD in general. This requires further analytical studies, but they are beyond the scope of this article (Moreno-Bote, R. et al., 2011).

In the literature, there have been other approaches for particle filtering without importance weights, derived rigorously from mathematical filtering theory. One of these approaches is the Ensemble Kalman Filter (EnKF). It is a generalization of the extended Kalman Filter in which copies of the Kalman filter are evolved with a gain that is computed using the empirical variance of the particle samples. As such, for a linear filtering problem, the EnKF outlined in the reference below, is equivalent to the NPF with empirical gain factor, but the two algorithms diverge for non-linear problems, where the EnKF has to resort to special techniques in order to remain stable (Legenstein, R. & Maass, W., 2014). The FBPF is based on a similar SDE for the particle trajectories as the one the author proposes in formula 4, i.e. it exhibits the general structure of the hidden SDE as well as novelty-based gain-feedback structure. The FBPF arises as the solution of an optimal control problem, and the optimal feedback gain is the solution of an Euler-Lagrange boundary-value problem (BVP). This BVP cannot be solved in closed form in the multidimensional case, but it can be approximated by a Galerkin method (Deneve, S. et al., 2007). It is noteworthy that the particle dynamics in the Ensemble Kalman-Bucy Filter (EnKBF) outlined in reference below, are identical to that of the FBPF employing this approximation. The NPF with empirically-adjusting gain \( W \) resamples the FBPF with Galerkin approximation, up to a slight modification in the novelty term, although the respective approaches to deriving both filters is fundamentally different. Thus far, the main difference between both filters is that a learning framework is provided with the NPF, which allows for learning of model parameters and filter gain factors, whereas the FBPF only accounts for filtering, and not for identification, problems (Crisan, D. & Xiong, J., 2010).

The author examined the problem of studying the parameters of the generative model, as well as tuning the gain matrix \( W \) using an interactive gradient rise along the logarithm of the probability of observation. For the class of models the author is considering, the logarithmic probability has a very simple representation in terms of the optimal filter. Replacing the optimal filter with an NPF particle estimate makes it possible to obtain completely recursive learning rules for the model parameters, as well as for the gain matrix. The author currently has no difficult results regarding the convergence of the algorithm. For hidden Markov models (HMMs) (discrete time and discrete state space), there are online logarithmic likelihood approaches (but no particles), and it has recently been shown that they converge under relatively mild assumptions. Similar ML online approaches with weighted particle filters (in discrete time) are possible, but they are negatively affected by particle degeneration due to weight reduction, and this is an open - but interesting - question, how would these results go over to the particle less method, For HMM as an alternative to online gradient methods is proposed online maximization of expectations (online EM), which is based on the well-known EM algorithm for offline learning. However, at present there are only partial results of convergence, and a generalization of online EM for continuous-time models has not been fully established. If such a generalization is possible in the future, it will be interesting to see if the problems of locality of gradient ascent online can be solved.

The question of how the brain can perform filtering was definitely decided earlier. In particular, filtering algorithms based on linear generating models have been the subject of extensive research and mainly study how the analytical solutions to this problem, the Kalman filter, can be implemented using neurons. However, the posterior result obtained as a result of the Kalman filter is always Gaussian, which strongly limits and does not adequately reflect the
activity distributions observed in neurons. A more general approach, which is not only suitable for a nonlinear generating model, but also includes the study of parameters. It is based on sample paths generated by the biologically probable dynamics of neurons, which are weighted in accordance with the logarithmic likelihood ratio for diffusion processes, and parameters studied with maximized expectations. An important difference of our approach is that both inference and training are carried out all the way, while NPF is an online algorithm. Another related approach, where neurons are considered Monte Carlo samplers in the hidden Markov model, offers both online (non-linear) filtering and training in a missing network. However, the neuronal representation of the latent state is significantly different from our approach. Indeed, it relies on the discretization of the state space, and therefore the author expects this approach to scale unfavorably as the latent dimension increases.

The structure of the neural network (Figure 1b), the author proposes to implement neural dynamics in accordance with the formula 4, structurally similar to that proposed by Rao and Ballard. As in their model, the author presents neural activity in terms of their instantaneous launch speed, which is an approximation to the peak nature of biological neurons. In their predictive coding model, the central role is given to the predictive error signal, which can be compared with the dynamics of novelty neurons or the novelty signal dt. In each subnet, Indeed, recent experimental data seem to confirm the existence of neurons that exhibit predictive-stimulating responses. Due to the similarities between predictive coding and NPF, the equations for neural dynamics and for determining the generative weight in the low noise limit are similar. However, our model generalizes the model of Rao and Ballard in the sense that the author allows a dynamic a priori, which is directly reflected in the dynamics of filtering neurons (Hennequin, G. et al., 2014).

It should be emphasized that NPFs are as indicated in the formula (4) ansatz. This means that other options are possible that include more likely biological features. Since the very concept of biological plausibility is not well defined, the point here is not to assert that one particular dynamics is more plausible than another, but to insist on the fact that the NPF does not depend on significance weights, and this ensures great advantage in terms of implementation (to be neural or silicon-based) (Kantas, N. et al., 2015).

The main limitation of this structure is that the training rules are usually not local, which is mainly due to the so-called filter derivative. The author saw that for a small observation noise, the training rules can be approximated to show the Hebbian and therefore local structure, but the author cannot expect the brain to rely only on the observation noise to be small. To overcome the non-locality in the training rules for more observational noise, another possibility is to consider a network with only one particle, which gives us an a posteriori in time if the sampler dynamics is fast enough (similar to the MCMC approach, where the back is motionless). The author could show that if the decoding weight W is studied, the filtering efficiency of a single-particle NPF is still reasonable, and may also be sufficient for training parameters. This approach can be further expanded by considering N independent, i.e. non-binding filtering networks. Another possibility to overcome non-locality in the rules of instruction may be to consider the learning process as a conclusion and impose a local ansatz on the rule of instruction.

Regardless of the limitations and the specific structure of the neural network, two central aspects of our work, namely the sampling-based representation and the adaptive amplification filtering algorithm, lead to the following consequences: (a) the neuron variability increases with the uncertainty of the signs, (b) the neuron dynamics is due to the prediction error, i.e. the discrepancy between the predicted observations and the actual observations, and (c) the network is resistant to neuron failure.

References

[16] Fiser, J., Berkes, P., Orban, G. & Lengyel, M., (2010), Statistically optimal perception and learning: from...


[36] International Journal of Applied Nanotechnology Research(IJANR), DOI: 10.4018/IJANR.2019010104


