Learning stochastic filtering

RAHUL O. RAMAKRISHNAN¹, ANDREA AUCONI¹ and BENJAMIN M. FRIEDRICH¹,²(a)

¹ cfad, Technische Universität Dresden - 01069 Dresden, Germany
² Cluster of Excellence “Physics of Life” - 01307 Dresden, Germany

received 22 June 2022; accepted in final form 24 October 2022
published online 17 November 2022

Abstract – We quantify the performance of approximations to stochastic filtering by the Kullback-Leibler divergence to the optimal Bayesian filter. Using a two-state Markov process that drives a Brownian measurement process as prototypical test case, we compare two stochastic filtering approximations: a static low-pass filter as baseline, and machine learning of Volterra expansions using nonlinear Vector Auto-Regression (nVAR). We highlight the crucial role of the chosen performance metric, and present two solutions to the specific challenge of predicting a likelihood bounded between 0 and 1.

Introduction. – Stochastic filtering is the estimation of the current state of a stochastic process based on a history of noisy measurements [1–4]. The optimal filter uses Bayes formula with the true measurement probabilities to continuously update a likelihood of states. If the stochastic dynamics of the underlying process is linear and measurement noise is Gaussian, this optimal filter is the celebrated Kalman filter [2]. For nonlinear problems, numerical approximations are available in the statistics literature [4]. These approximations become especially important if the true measurement probabilities are not known. However, approximations often lack interpretability and it may be difficult to evaluate their performance.

Given a generic stochastic filtering scheme, the uncertainty of its estimates can be decomposed into two contributions: the entropy of the optimal filter, and the Kullback-Leibler (KL) divergence [5,6] from the optimal filter. This second term was already suggested as a general measure of model quality in [7]; here, we propose it as a performance measure for stochastic filtering approximations that track a time-dependent probability distribution.

From an engineering point of view, stochastic filtering can be viewed as a problem of system identification: find an approximative model of a dynamical system from its time series [8,9]. A priori knowledge about the system allows to make an informed guess, i.e., a heuristic model with few parameters that can be tuned to an optimum. In the absence of such knowledge, one may harness machine learning to learn “black-box” approximation with many parameters [10,11]. We will demonstrate the applicability of both approaches on a test problem, and highlight the specific challenges of predicting a probability that is bounded between zero and one, as well as possible solutions.

The manuscript is organized as follows: After briefly introducing stochastic filtering, we review the Kullback-Leibler divergence as an insightful performance measure for model quality. We set up a minimal model of noise-corrupted measurements of a bistable process, which is sufficient to highlight key differences between a linear low-pass filter (approximation I), and a nonlinear Volterra expansion of the optimal filter (approximation II).

Stochastic filtering. – We start by reviewing the basic features of stochastic filtering. Let \( x_j \) be the unknown state of a stochastic dynamical system at discrete time \( j \), and \( m_j \) a (cumulative) measurement process, whose stochastic increments \( \Delta m_j = m_j - m_{j-1} \) depend on the state \( x_j \) of the hidden process. The likelihood \( p(x_j) \) formalizes the knowledge at time \( t_j \) about the current state of the hidden process \( x_j \) (conditional to all past measurements and possibly an initial prior \( p(x_0) \)).

The likelihood evolution is decomposed into a prediction step, \( p(x_{j+1}) = \int dx_j p(x_{j+1}|x_j)p(x_j) \), which replicates the stochastic dynamics \( p(x_{j+1}|x_j) \) of the
hidden process, and an update step, \( p(x_{j+1} | \Delta m_{j+1}) = p(\Delta m_{j+1} | x_{j+1}) p(x_{j+1}) / p(\Delta m_{j+1}) \), which uses Bayes formula to incorporate the latest measurement \( \Delta m_{j+1} \). Here, the measurement probability \( p(\Delta m_{j+1}) = \int dx_{j+1} p(\Delta m_{j+1} | x_{j+1}) p(x_{j+1}) \) acts as normalization factor. If \( p(\Delta m_{j+1} | x_{j+1}) \) is the true measurement probability, the filter \( p \) is optimal. The posterior \( p(x_{j+1} | \Delta m_{j+1}) \) serves as new prior for the next time step \( j + 2 \). In the limit of a vanishing discretization, this procedure yields a time-continuous version of stochastic filtering. Below, we will assume that the dynamics of the hidden process is stationary, and that the influence of the initial prior vanishes in the long-time limit.

**Model quality.** In many application cases, one has to resort to approximations of optimal stochastic filtering. The measurement process may not be known, or agents may lack the elaborate machinery needed for optimal filtering, as common for biological cells performing chemical sensing [12,13]. In these cases, we have a sub-optimal filter, whose likelihood estimates \( q(x_j) \) will in general differ from those of the optimal filter, \( q(x_j) \neq p(x_j) \). The performance of this filter \( q(x_j) \) can be sub-optimal due to reduced dimensionality, intrinsic noise, or systematic errors. To quantify differences in performance, we examine relative entropies. The *stochastic entropy* for the optimal filter, \( s(x) = -\ln p(x) \), is the negative log-likelihood of the actual state \( x \). Its ensemble average defines the macroscopic entropy, \( S[p] = \langle s(x) \rangle_p = \int dx p(x) s(x) \), which quantifies the current uncertainty of the optimal filter. The stochastic entropy can also be defined for the sub-optimal estimate, \( s^q(x) = -\ln q(x) \). Its ensemble average with respect to the true density is the cross-entropy, \( H[p,q] = \langle s^q(x) \rangle_p = \int dx p(x) s^q(x) \), which quantifies the current uncertainty of the sub-optimal filter. In general, \( H[p,q] \) is different from the macroscopic entropy \( S[q] = \langle s^q(x) \rangle_q \) of \( q(x) \), because the sub-optimal filter has an erroneous estimate of its own uncertainty.

Optimal and sub-optimal estimates are related by a fluctuation theorem, \( \langle \exp[s(x) - s^q(x)] \rangle_p = 1 \), which follows from the normalization condition for \( q(x) \). Using Jensen’s inequality, we obtain the standard upper bound on the entropy by the cross-entropy, \( S[p] \leq H[p,q] \), with equality only if \( p = q \). This means that the log-likelihood of the actual state evaluated by the optimal filter is on average equal to or greater than the log-likelihood of any other scheme, which we would indeed name sub-optimal. Still, the fluctuation theorem allows that in some realizations one can observe apparent violations \( p(x) < q(x) \).

The excess uncertainty of the sub-optimal filter is the Kullback-Leibler divergence \( D_{KL}[p||q] = H[p,q] - S[p] \), which gives \( D_{KL}[p||q] = \ln(p/q) \). The Kullback-Leibler divergence was already proposed to measure model quality in [7]. Here, we propose to use a time average \( \langle D_{KL}[p||q] \rangle_t \) to quantify the performance of sub-optimal filters.

Below, we will consider a discrete state space for the hidden process \( x \) composed of just two states \( x = \pm 1 \). By slight abuse of notation, we can identify \( p \) with the scalar \( p(x = 1) \), and similarly \( q \) with \( q(x = 1) \), and write

\[
D_{KL}[p||q] = p \ln \left( \frac{p}{q} \right) + (1-p) \ln \left( \frac{1-p}{1-q} \right) .
\]

Next, we introduce a minimal bistable model of stochastic filtering, and apply eq. (1) to quantify the performance of sub-optimal approximations.

**Bistable model.** Consider a continuous-time Markov process \( x \) that jumps with symmetric rate \( r > 0 \) between two states, \( x \in \{-1, +1\} \), see fig. 1(a). We assume that we can measure this hidden state \( x_t \) through a time-continuous measurement process \( m_t \) described by the following stochastic differential equation (SDE):

\[
dm = \gamma xdt + \sqrt{2D}dW,
\]

where \( \gamma > 0 \) is the signal strength and \( \sqrt{2D}dW \) denotes Gaussian white noise with noise strength \( D \). The measurement model eq. (2) describes, e.g., activation of receptors in biological cells exposed to a time-varying, bistable ligand-concentration \( x \) with stochastic ligand-receptor interactions subsumed as measurement noise [13–15].

We denote by \( p \) the likelihood of state \( x = 1 \). The optimal Bayesian estimation scheme allows to derive evolution equations for quantities of interest, such as the time-dependent likelihood \( p(x,t) \), or the expected entropy variation \( \langle dS \rangle / dt \). The calculation methods are standard and can be found, e.g., in [13,14] (likelihood evolution), and [16] (entropy variation).

We report analytical results for the specific application problem embodied in eq. (3); calculation details are provided in the Supplementary Material [Supplementarymaterial.pdf](SM)

\[
dp = \frac{\gamma}{D} p(1-p) (dm - \langle dm \rangle_p) + r(1-2p)dt .
\]

Here, \( \langle dm \rangle_p = \gamma (2p - 1)dt \), and products with the measurement process \( dm \) are to be interpreted in the Itô scheme of non-anticipative stochastic calculus [1,17]. Although the measurement model eq. (2) is linear, the fact that we consider a likelihood distribution on a discrete set (here: \( \{-1, +1\} \)) results in a genuinely nonlinear evolution equation eq. (3) for Bayesian updating. We will see below how this nonlinearity prompts higher-order nonlinearities in filtering approximations.

Equation (3) also implies an evolution equation for the entropy \( S[p] \), and, in particular, for the expected entropy variation with respect to the current likelihood

\[
\frac{dS}{dt} = r(1-2p) \ln \left( \frac{1-p}{p} \right) - \frac{\gamma^2}{D} p(1-p) .
\]

The first term in eq. (4) describes the expected increase of entropy in the prediction step, while the second term describes the expected decrease of entropy in the update
Learning stochastic filtering

Fig. 1: (a) As prototypical stochastic filtering task, we consider a Markov process \( x(t) \) jumping between two states \(+1\) (upper panel, green shading) and \(-1\) (no shading) that drives a noisy measurement process \( dm \), see eq. (2). Bayesian inference yields an optimal estimate of the likelihood \( p(t) \) of \( x(t) = +1 \) (lower panel, red curve), see eq. (3). As a sub-optimal approximation \( q(t) \), we consider a low-pass filter (blue curve), eq. (5). Histogram of these likelihood estimates (with invariant density \( \rho(p) \)) are shown to the right. (b) Mean Kullback-Leibler divergence \( \langle D_{KL}[p||q] \rangle \) between optimal and sub-optimal filter as a function of the inverse relaxation time \( \beta \) of the sub-optimal low-pass filter for different noise strengths \( \gamma \) and switching rates \( r \) of \( x(t) \) (where \( \beta, \gamma, \) and \( r \) are measured relative to \( 2D = 1 \)). (c) There exists an optimal inverse time-scale \( \beta^* \) minimizing \( \langle D_{KL}[p||q] \rangle \), which increases with \( r \). Results are shown for two values of \( \gamma \), corresponding to low signal strength (\( \gamma = 0.1 \), solid), and high signal strength (\( \gamma = 1 \), dashed). Error bars given by standard error of the mean (\( n = 5 \) realizations) are virtually invisible.

step. Both terms depend on \( p \) in a nonlinear fashion, and are invariant under the symmetry operation \( p \rightarrow 1 - p \). The prefactor \( \gamma^2/D \) of the second term can be interpreted as a rate constant of information gain, similar to \( \gamma \), \( D \). The thermodynamic interpretation of entropy variations in stochastic systems with measurement components has been discussed in [20–23].

Approximation I: low-pass filter. – To construct our first approximation, consider a low-pass filter of the measurement process, \( d\xi = dm - \beta\xi dt \), with inverse relaxation time \( \beta \). We want to estimate a likelihood \( q(\xi) \) for \( x_t = +1 \) based on \( \xi_t \). In the limit of rare switching \( r \ll \beta \), the probability density \( p(\xi|x) \) is given by the steady-state density of an Ornstein-Uhlenbeck process: a normal distribution with mean \( \gamma x/\beta \) and variance \( D/\beta \). By treating the hidden state as static, we introduce a systematic bias. We can now apply Bayes formula using this steady-state density and the prior \( p(x = +1) = 1/2 \), and obtain the approximated likelihood

\[
q\left(x_t = +1|\xi_t\right) = \frac{1}{1 + \exp(-2\gamma \xi_t/D)},
\]

which describes a logistic curve. Applying Itô’s Lemma [1, 17] to eq. (5), we obtain an evolution equation for the approximated likelihood

\[
\frac{dq}{D} = \frac{2\gamma}{D}q(1-q)(dm - 2(dm)_q)
- \beta q(1-q)\ln\left(\frac{q}{1-q}\right)dt,
\]

where \( \langle dm\rangle_q = \gamma(2q-1)dt \) is the expectation value of the measurement increment according to the approximated likelihood \( q \).

We numerically determined the performance of this approximative stochastic filter using the Kullback-Leibler divergence, see fig. 1(b). Higher signal strengths \( \gamma \) reduce the performance relative to the optimal filter (higher \( D_{KL} \)), because likelihoods close to zero and one become more frequent, which are more difficult to approximate. Similarly, a lower switching rate \( r \) makes the estimation problem easier for both the sub-optimal and the optimal filter, but \( D_{KL}[q||p] \) may nonetheless increase, as only the optimal filter has the correct functional dependence on the measurement process. Lastly, the inverse relaxation time \( \beta^* \) of the low-pass filter marks a trade-off between responding fast (small \( \beta \)) or responding precisely (high \( \beta \)), resulting in an optimum \( \beta^* \) that minimizes the Kullback-Leibler divergence. This optimal \( \beta^* \) increases with the switching rate \( r \) as expected, with approximately linear scaling for low \( \gamma \), see fig. 1(c).

In principle, an extended Kalman filter (EKF) could be used to estimate \( p(x) \), see SM for details. It turns out that this EKF with a multiplicative noise term can become trapped near \( x \approx 0 \), except for large signal strength \( \gamma^2 \gg Dr \). This EKF (with simple resetting in case of trapping) can predict likelihoods better than the optimal low-pass filter, but is outperformed by nonlinear vector auto-regression (nVAR), as discussed next.

Approximation II: machine learning. – As an alternative approximation of stochastic filtering, we consider nonlinear vector auto-regression (nVAR) [24], a state-of-the-art machine learning approach that learns, e.g., Volterra expansions from time series data. After briefly reviewing the theory behind nVAR, we propose how nVAR can be adapted for the problem of stochastic filtering, and demonstrate its performance for this task.
Fig. 2: (a) nVAR is a method to approximate arbitrary dynamic relationships between a time-dependent input \( u(t) \) and an output \( y(t) \) by learning the kernels of a truncated Volterra expansion, eq. (8). During training, for each time point \( t_j \), feature vectors \( \mathbf{u}_{n,j} \) are constructed from monomials of order \( n \) in the time-delayed inputs \( u(t_j), u(t_{j-1}), \ldots, u(t_{j-\alpha}) \). These feature vectors (up to maximal order \( n \leq N \)) are combined into a single feature matrix \( \mathbf{U} \). Linear regression with regularization (ridge regression), eq. (10), yields an optimal weight vector \( \mathbf{w} \) such that the vector \( y \) of outputs \( y(t_j) \) is approximated as \( y \approx \mathbf{w} \cdot \mathbf{U} \), i.e., \( y(t_j) \approx \mathbf{w} \cdot \mathbf{u}_j \) for all \( j \). (b) Application of nVAR to the stochastic filtering problem of fig. 1(a) with \( N = 3 \), equivalent to learning the kernels \( h_1, h_2 \) and \( h_3 \) of a Volterra expansion; \( h_3(\tau) \) is shown (blue curve, average of 10 realizations). Approximately, \( h_3(\tau) \approx A e^{-\beta \tau} \) with fit parameters \( A \approx 3/2 \) and \( \beta \approx 40/3 \) (red curve in inset). Parameters: \( \gamma = 3 \text{s}^{-1}, \tau = 3 \text{s}^{-1}, \beta = 0.5 \text{s}^{-1} \), duration of training data 800 s, \( \Delta t = 0.01 \text{s} \), delay \( k = 40 \), \( N = 3 \), ridge parameter \( \alpha = 10^{-3} \).

Generally, any dynamical system with input \( u(t) \) and output \( y(t) \) can be thought of as a functional relationship \( y(t) = \mathcal{H}[u(t)] \) for some functional \( \mathcal{H} \) that operates on time-dependent functions. For linear and time-invariant systems, one can show that this input-output relationship is just a convolution,

\[
y(t) = h_0 + \int_{-\infty}^{\infty} h_1(\tau) u(t - \tau) d\tau,
\]

with a constant \( h_0 \) and a suitable kernel \( h_1(\tau) \) (sometimes called linear response function or susceptibility). Causality implies \( h_1(\tau) = 0 \) for \( \tau < 0 \). For nonlinear systems, this generalizes to the Volterra series expansion, a well-known description in system identification [25],

\[
y(t) = H_0[u(t)] + H_1[u(t)] + H_2[u(t)] + \ldots + H_n[u(t)] + \ldots
\]

(8)

Here, \( H_0[u(t)] = h_0 \), and \( H_n \) is the \( n \)-th order Volterra operator, which may be written as a “higher-order convolution”,

\[
H_n[u(t)] = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \ldots, \tau_n) \\
\cdot u(t - \tau_1) \ldots u(t - \tau_n) \, d\tau_1 \ldots d\tau_n.
\]

Theoretically, inputs from a distant past and higher-order kernels influence the output \( y(t) \). For practical purposes, however, we can only use a finite input history and must truncate the Volterra expansion at some order. Despite these limitations, learning Volterra kernels using nVAR could approximate even non-polynomial systems surprisingly well [24].

Figure 2 outlines the nVAR algorithm: The training data consists of discrete input and output time series \( \{u(t_j)\} \) and \( \{y(t_j)\} \). For each time point \( t_j = j \Delta t \), a feature vector \( \mathbf{u}_j \) is constructed, whose components comprise all unique monomials (up to a given order \( N \)) built from the \( k \) + 1 delay terms \( u(t_j), u(t_{j-1}), \ldots, u(t_{j-k}) \). The computational cost scales as \( O(k^N) \). The feature vectors \( \mathbf{u}_j \) are then concatenated into a matrix \( \mathbf{U} \), which, in direct analogy to the Volterra expansion, should be linearly related to the output vector \( \mathbf{y}_{\text{target}} \) (whose components are the \( y(t_j) \)) by a yet unknown weight vector \( \mathbf{w} \) as \( \mathbf{w} \cdot \mathbf{U} \approx \mathbf{y}_{\text{target}} \). One can think of \( \mathbf{w} \cdot \mathbf{U} \) as approximating eq. (8) by keeping only terms of order \( n \leq N \), cutting-off integration limits as \( 0 \leq \tau \leq k \Delta t \) in eq. (9), and discretizing time with timestep \( \Delta t \). The optimal weights \( \mathbf{w} \) are found by the minimization

\[
\| \mathbf{w} \cdot \mathbf{U} - \mathbf{y}_{\text{target}} \|^2 + \alpha \| \mathbf{w} \|^2 \rightarrow \min,
\]

(10)

where \( \| \cdot \| \) stands for the Euclidean norm. Equation (10) includes a penalty for large weights with so-called ridge parameter \( \alpha \) to reduce over-fitting. The optimization problem eq. (10) represents a standard task of quadratic programming, and can be efficiently solved by reduction to a linear-algebra problem with formal solution [26],

\[
\mathbf{w} = \mathbf{y}_{\text{target}} \mathbf{U}^T (\mathbf{U} \mathbf{U}^T + \alpha \mathbf{I})^{-1}.
\]

(11)

After training, the performance of the nVAR machine can be evaluated using an independent test data set: Multiplying the learned weight vector with the new feature matrix constructed from the test input yields a predicted
Fig. 3: Performance of different approximations for stochastic filtering. (a) Performance measured in terms of the mean-squared error (mse) between true and predicted likelihood. LPF: mse for approximation I using the low-pass filter eq. (5). We used the optimal inverse relaxation time $\beta^*$ that minimizes either mse (hatched green), or the Kullback-Leibler divergence (as considered in panel (b); pale green). Clipping: mse for likelihoods predicted using nVAR for different truncation orders of the Volterra expansion ($h_1$: $N = 1$, $h_1 + h_2$: $N = 2$, $h_1 + h_2 + h_3$: $N = 3$). To ensure that likelihoods are bounded within $[0,1]$, we clipped predicted likelihoods above 1 and below 0 (blue). Logit: as an alternative method to ensure that likelihoods lie within $[0,1]$, we applied a nonlinear logit transformation, eq. (12), to the likelihoods of the training data before training; the inverse transformation was then applied after prediction, but before the mse was computed (red). (b) Same as panel (a), but using mean Kullback-Leibler divergence ($D_{KL}$) as performance measure. Here, we use that the likelihood $p = p(x = \pm 1)$ fully determines the likelihood distribution $p(x)$ for $x = \pm 1$ and thus the Kullback-Leibler divergence, eq. (1). Parameters: $\gamma = 3s^{-1}$, $r = 3s^{-1}$, $D = 0.5s^{-1}$, duration of training time series 800s, duration of test time series 400s, $\Delta t = 0.01s$, delay $k = 40$, ridge parameter $\alpha = 10^{-1};$ reported results represent mean ± s.e.m. from 10 realizations.

A peculiarity of stochastic filtering is that likelihoods must always lie in the interval $[0,1]$. Per se, the nVAR algorithm does not respect this property. To solve this issue, we suggest two different solutions. The first, simple solution is to chop the predicted output above 1 and below 0. This makes computed mean-squared errors more meaningful, and is even a prerequisite to compute a Kullback-Leibler divergence.

As a second solution, we applied a nonlinear logit transformation to the likelihoods $p$,

$$\phi = \logit(p) = \ln \frac{p}{1-p},$$

(12)

which maps $0 < p < 1$ bijectively to $-\infty < \phi < \infty$. We then use nVAR to predict $\phi(t_j)$ with input $\Delta m_j$; the predicted output $\hat{\phi}$ is then transformed back to the corresponding likelihood $\hat{p}$ using the inverse transformation

$$\hat{p} = \logit^{-1}(\hat{\phi}) = [1 + \exp(\hat{\phi})]^{-1}.$$  

(13)

It is worth noting the formal similarity between eq. (13) and the simplified approximation eq. (5). On a practical note, to avoid large values of $\phi$, which may cause problems during learning, we clipped inputs $p$ to the interval $[\epsilon, 1-\epsilon]$ for a small number $\epsilon = 10^{-8}$ before computing the transformed input $\phi$.

Figure 3 compares these two approaches to deal with the fact that likelihoods are bounded, using two different performance measures, the conventional mean-squared error (mse), and the mean Kullback-Leibler divergence. Generally, the first approach (clipping) performs slightly better than the second approach (logit) in terms of the mean-squared error, which is expected because the first approach directly minimizes this measure. In contrast,
the logit approach is superior when the mean Kullback-Leibler divergence should be minimized. This observation can be rationalized as follows: the nonlinear logit transformation eq. (12) “stretches” likelihoods close to 0 or 1 out; as the Kullback-Leibler divergence is rather sensitive to these likelihoods, the logit approach generally performs better in terms of the Kullback-Leibler divergence. As discussed above, including second-order terms does not improve performance for any of the two approaches as expected by symmetry, whereas third-order terms improve performance. This improvement is stronger for the clipping-approach compared to the logit approach, presumably, because the nonlinear logit transformation partially accounts already for the nonlinear dependence of \( p \) on the input \( dm \). These observations were confirmed for various parameter choices (not shown). In the limit of strong noise (small \( \gamma \)), likelihoods \( p \) will deviate only little from 1/2, rendering the dependence of \( p \) on \( dm \) approximately linear, which reduces the benefit of including third-order terms. Remarkably, the clipping approach using only linear terms \( (h_1) \) outperforms the simple low-pass filter even if an optimal inverse relaxation time is chosen for the later. This highlights the short-comings of the indirect approach of approximation I, which first computes the auxiliary variable \( \xi \), which is then converted into an estimate for the likelihood using the steady-state probability distribution, eq. (5), instead of learning likelihoods directly as in approximation II.

For machine learning tasks, it is pivotal to choose hyper-parameters judiciously. The number \( k \) of delay steps should satisfy \( k > (\beta^* dt)^{-1} \) to cover the time-window, where the expected Volterra kernels \( h_1(\tau) \sim \exp(-\beta^* \tau) \) are large. Figure 1(c) suggests \( \beta^* \sim r \), where \( r \) is the switching rate of the two-state Markov process.

Next, under-fitting or over-fitting will occur when the number of weights and thus fit parameters is either much smaller or much bigger than the size of the training data, respectively. Intriguingly, performance is worst when the number of weights exactly matches the size of the training data [27]. If the number of weights increases further and exceeds the number of training data points, learned weights become partially randomized, which reduces the adverse effect of over-fitting [28], see also the SM. For our application example, performance was best when the number of weights was in a range of 10–30% of the number of data points of the training data.

The common problem of over-fitting and resultant abnormal weights caused by noise in finite-size input data can be effectively mitigated by means of regularization, i.e., introducing a regularization term with ridge parameter \( \alpha \) in the optimization problem equation (11). Prediction performance for various choices of \( \alpha \) are shown in the SM. Interestingly, the logit approach of applying a nonlinear transformation to the output before learning is rather sensitive to the choice of \( \alpha \) if third-order terms are included, whereas the performance of the simpler clipping approach is robust and virtually independent of \( \alpha \).

**Conclusion.** — We addressed the general problem of stochastic filtering to infer a time-dependent hidden state from noisy measurements. Because the optimal filter based on Bayesian inference can be analytically calculated only for simple examples, efficient approximations are needed. Here, we propose nonlinear vector auto-regression (nVAR) [24] as an efficient method to learn the dynamic relationship between the likelihood of hidden states and noisy, time-dependent input. Specifically, nVAR allows to learn a Volterra expansion relating input and output, in our case, noisy measurements as input and likelihoods of hidden states as output. As a baseline, we additionally consider a simple low-pass filter. We quantify the performance of these approximations of stochastic filtering in terms of the Kullback-Leibler divergence from the optimal filter.

Using higher-order terms in the Volterra expansion or longer delays in nVAR increases the accuracy of the predictor, but also increases computational cost. We show that nVAR with reasonable delays and only linear terms already performs better than the simple low-pass filter (and also better than an extended Kalman filter). Including third-order terms slightly improves performance further, whereas second-order terms are dispensable because of the symmetry of the problem.

A key issue is that likelihoods must always be bounded between zero and one—a property not automatically respected by common approximation schemes. We propose two possible solutions to this specific challenge of stochastic filtering: i) clipping of predicted likelihoods outside the admissible range, and ii) applying a nonlinear logit transformation before training, and back-transformation of predicted outputs. The second approach displays higher fidelity when it comes to predicting likelihoods close to zero or one, and, concomitantly, shows a lower mean Kullback-Leibler divergence than the first approach. At the same time, the mean-squared error resulting from the second approach is still acceptable. As a drawback, the second approach is sensitive to proper regularization, and requires judicious choice of a ridge parameter. Potentially, nonlinear models may provide a solution, e.g., using a sigmoidal function applied to a weighted sum of input features. However, learning nonlinear models would be considerably harder than nVAR and convergence to a global optimum could not be guaranteed.

In conclusion, the logit transformation seems a viable approach that minimizes the mean Kullback-Leibler divergence, while the mean-squared error remains acceptable. As a caveat, the logit transformation is more susceptible to over-fitting and requires suitable regularization, while the simpler clipping approach is more robust.

While we restricted ourselves in nVAR to learning nonlinear terms up to third order, the method could be applied up to arbitrary order or using longer delays, with the availability of sufficient training data and computational resources being the only bottle-necks. Computational complexity could be reduced by careful design of
the feature vector, e.g., using coarser time-sampling for inputs with longer delays, or problem-specific basis functions constructed from the inputs [10]. Lasso regression allows to shrink less important weights to zero [29]. As an alternative to nVAR, multilayer neural networks have been proposed to learn Volterra expansions, with a direct relation between the internal weights of the network and Volterra kernels [11,30,31]. We have shown how learning Volterra expansions can be adapted to find stochastic filtering approximations, enabling future applications for on-line decision making.

***

ROR was supported by the Ministry of Science and Art of the Federal State of Saxony, Germany (Forschungsprojektförderung Titelgruppe 70 des Sächsischen Staatsministerium für Wissenschaft und Kunst) through grant 100400118 to MARC TIMMÉ and BMF. AA was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through grant FR3429/3-1 to BMF. BMF is supported by the DFG through Excellence Strategy - EXC-2068 - 390729961. ROR, AA, and BMF acknowledge support through the Center for Advancing Electronics Dresden (cfaed).

Data availability statement: All data that support the findings of this study are included within the article (and any supplementary files).

REFERENCES

[1] Rogers L. C. G. and Williams D., Diffusions, Markov Processes and Martingales: Volume 2, Itô Calculus (Cambridge University Press) 2000.
[2] Kalman R. E., J. Basic Eng., 82 (1960) 35.
[3] Fuiisaki M., Kallianpur G. and Kunita H., Osaka J. Math., 9 (1972) 19.
[4] Petris G., Petrone S. and Campagnoli P., Dynamic Linear Models, in Dynamic Linear Models with R (Springer) 2009, pp. 31–84.
[5] Amari S. I., Information Geometry and its Applications, Vol. 194 (Springer) 2016.
[6] Cover Thomas M., Elements of Information Theory (John Wiley & Son) 1999.
[7] Bialek William, Palmer Stephanie E. and Schwab David J., What makes it possible to learn probability distributions in the natural world?, arXiv:2008.1229v2 (2020).
[8] Wang Wen-Xu, Lai Ying-Cheng and Grebogi Celso, Phys. Rep., 644 (2016) 1.
[9] AlRahman AlMosani Abd, Sun Jie and Bolltt Erik, Chaos, 30 (2020) 013107.
[10] Brunton S. L., Proctor J. L. and Kutz J. N., Proc. Natl. Acad. Sci. U.S.A., 113 (2016) 3932.
[11] Cheng C. M. et al., Mech. Syst. Signal Process., 87 (2017) 340.
[12] Bialek William, Biophysics: Searching for Principles (Princeton University Press) 2012.
[13] Kobayashi Tetsuya J., Phys. Rev. Lett., 104 (2010) 228104.
[14] Sigglia Eric D. and Vergassola Massimo, Proc. Natl. Acad. Sci. U.S.A., 110 (2013) E3704.
[15] Kobayashi T. J., Phys. Rev. Lett., 106 (2011) 228101.
[16] Auconi A., Novak M. and Friedrich B. M., EPL, 138 (2022) 12001.
[17] Karatzas Ioannis and Shreve Steven, Brownian Motion and Stochastic Calculus, Vol. 113 (Springer Science & Business Media) 2012.
[18] Mora T. and Nemenman I., Phys. Rev. Lett., 123 (2019) 198101.
[19] Novak M. and Friedrich B. M., New J. Phys., 23 (2021) 043026.
[20] Sagawa T. and Ueda M., Phys. Rev. E, 85 (2012) 021104.
[21] Ito S. and Sagawa T., Phys. Rev. Lett., 111 (2013) 180603.
[22] Bartolotta A., Carroll S. M., Leichenauer S. and Pollack J., Phys. Rev. E, 94 (2016) 022102.
[23] Auconi A., Giansanti A. and Klipp E., Entropy, 21 (2019) 177.
[24] GAUTHIER D. J. et al., Nat. Commun., 12 (2021) 1.
[25] Franz M. O. and Schölkopf B., Neural Comput., 18 (2006) 3097.
[26] Hoerl Roger W., Technometrics, 62 (2020) 420.
[27] Loog M. et al., Proc. Natl. Acad. Sci. U.S.A., 117 (2020) 10625.
[28] Belkin M. et al., Proc. Natl. Acad. Sci. U.S.A., 116 (2019) 15849.
[29] Tshimirani Robert, J. R. Stat. Soc. B, 58 (1996) 267.
[30] Wray J. and Green G., Biol. Cybern., 71 (1994) 187.
[31] PARKER R. E. and TUMMALA M., Identification of Volterra systems with a polynomial neural network, in ICASSP, Vol. 4 (IEEE Computer Society) 1992, pp. 561–564.