

Learning Optimal Filters Using Variational Inference

Enoch Luk^{*1} Eviatar Bach^{*123} Ricardo Baptista¹ Andrew Stuart¹

Abstract

Filtering—the task of estimating the conditional distribution of states of a dynamical system given partial, noisy, observations—is important in many areas of science and engineering, including weather and climate prediction. However, the filtering distribution is generally intractable to obtain for high-dimensional, nonlinear systems. Filters used in practice, such as the ensemble Kalman filter (EnKF), are biased for nonlinear systems and have numerous tuning parameters. Here, we present a framework for learning a parameterized analysis map—the map that takes a forecast distribution and observations to the filtering distribution—using variational inference. We show that this methodology can be used to learn gain matrices for filtering linear and nonlinear dynamical systems, as well as inflation and localization parameters for an EnKF. Future work will apply this framework to learn new filtering algorithms.

1. Introduction

Data assimilation (DA), the problem of estimating the states of a dynamical system, given partial and noisy observations, is ubiquitous in many fields. In particular, DA is essential in the numerical prediction of weather and climate (Kalnay, 2002; Carrassi et al., 2018). We briefly review the probabilistic formulation of the filtering problem; more details can be found in Jazwinski (1970); Law et al. (2015); Reich & Cotter (2015); Sanz-Alonso et al. (2023).

^{*}Equal contribution ¹Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, California, USA ²Department of Environmental Science and Engineering, California Institute of Technology, Pasadena, California, USA ³Department of Meteorology and Department of Mathematics and Statistics, University of Reading, Reading, UK. Correspondence to: Eviatar Bach <eviatarbach@protonmail.com>.

Workshop on Machine Learning for Earth System Modeling, 41st International Conference on Machine Learning, Vienna, Austria. Copyright 2024 by the author(s).

Consider the dynamics model for the state $v_j^\dagger \in \mathbb{R}^d$ as

$$v_{j+1}^\dagger = \Psi(v_j^\dagger) + \xi_j^\dagger, \quad j \in \mathbb{Z}^+ \quad (1a)$$

$$v_0^\dagger \sim \mathcal{N}(m_0, C_0) \quad \xi_j^\dagger \sim \mathcal{N}(0, \Sigma) \text{ i.i.d.}, \quad (1b)$$

where we assume that the sequence $\{\xi_j^\dagger\}_{j \in \mathbb{Z}^+}$ is independent of the initial condition v_0^\dagger . The observations $y_j^\dagger \in \mathbb{R}^p$ are given by

$$y_{j+1}^\dagger = h(v_{j+1}^\dagger) + \eta_{j+1}^\dagger, \quad j \in \mathbb{Z}^+, \quad (2a)$$

$$\eta_j^\dagger \sim \mathcal{N}(0, \Gamma) \text{ i.i.d.}, \quad (2b)$$

where we assume that the sequence $\{\eta_j^\dagger\}_{j \in \mathbb{N}}$ is independent of v_0^\dagger for all j , and that the two sequences $\{\xi_j^\dagger\}_{j \in \mathbb{Z}^+}$ and $\{\eta_j^\dagger\}_{j \in \mathbb{N}}$ are independent of one another.

We define, for a given and fixed integer J and any $j \leq J$,

$$V^\dagger = \{v_0^\dagger, \dots, v_J^\dagger\}, \quad Y_j^\dagger = \{y_1^\dagger, \dots, y_j^\dagger\}.$$

The aim of the *filtering* problem is to obtain the distribution $\Pi_j(v_j) = \mathbb{P}(v_j^\dagger | Y_j^\dagger)$ for any $j \leq J$. It can be shown that the solution to the filtering problem is given by operating on probability measures according to the following recursion, initialized at $\Pi_0 = \mathcal{N}(m_0, C_0)$:

$$\textbf{Prediction step:} \quad \widehat{\Pi}_{j+1} = P \Pi_j, \quad (3)$$

$$\textbf{Analysis step:} \quad \Pi_{j+1} = A(\widehat{\Pi}_{j+1}; y_{j+1}^\dagger).$$

We refer to P as the *prediction operator* and A as the *analysis operator*; the map P , viewed as taking the space of probability measures on \mathbb{R}^d , $\mathcal{P}(\mathbb{R}^d)$, into itself, is independent of time, whereas map A depends on time, although only through the observation y_{j+1}^\dagger . The linear prediction operator P acts on a measure π according to

$$P\pi(u) \propto \int_{\mathbb{R}^d} \exp\left(-\frac{1}{2}\|u - \Psi(v)\|_\Sigma^2\right) \pi(v) dv, \quad (4)$$

where $\|\cdot\|_\Sigma^2 = (\cdot)^\top \Sigma^{-1}(\cdot)$; we note that time-dependent generalizations are possible. The nonlinear analysis operator $A(\cdot; y^\dagger)$ acts on a measure π according to

$$A(\pi; y^\dagger)(u) = \frac{\nu_\pi(u, y^\dagger)}{\int_{\mathbb{R}^d} \nu_\pi(u, y^\dagger) du}, \quad (5)$$

where

$$\nu_\pi(u, y) \propto \exp\left(-\frac{1}{2}\|y - h(u)\|_\Gamma^2\right) \pi(u).$$

The analysis step thus multiplies by the likelihood $\mathbb{P}(y_{j+1}^\dagger | v_{j+1})$ and normalizes to make the resulting measure integrate to 1. Thus, in (3), the analysis step corresponds to an application of Bayes' Theorem with the prior $\widehat{\Pi}_{j+1}$, often referred to as the forecast distribution.

1.1. Motivation

Obtaining the filtering distribution is often intractable for high-dimensional and nonlinear systems. In the case that Ψ and h are linear, the filtering distribution is given in closed form by the Kalman filter. In the nonlinear case, the filter can be approximated by particle-based methods such as the particle filter and the ensemble Kalman filter. While the particle filter is asymptotically consistent with the true filter, it often performs poorly in high dimensions (Snyder et al., 2008). The ensemble Kalman filter (EnKF; see Appendix D) is often used in practical settings, but is known not to be consistent with the true filter in the nonlinear case (Le Gland et al., 2011; Carrillo et al., 2024). Moreover, it requires tuning of parameters such as inflation and localization to perform well in practice (Carrassi et al., 2018).

Previous work has focused on learning filtering densities within a parameterized class (see literature review below). Instead, we assume that the prediction step is known and learn parameters in the analysis map that takes samples from the forecast to the filtering distribution. Such a parameterized analysis step may arise, for instance, from using an EnKF with unknown tuning parameters, or by taking the analysis step to be a neural network with unknown weights. This analysis map leads to an analysis operator A_θ . Here, we present an approach for learning parameters θ^* so that the parameterized filter is close to the true one, i.e., $A_{\theta^*} \approx A$.

1.2. Literature Review and Contributions

A framework for jointly learning the forecast and analysis step was proposed by Boudier et al. (2023). Here, we assume a known forecast model and only focus on the analysis step. Other approaches for variational filtering, without a focus on learning parameterized analysis maps, have been presented in Marino et al. (2018); Campbell et al. (2021).

In our numerical experiments, we focus on learning a parameterized gain as part of the analysis step to filter linear and nonlinear dynamical systems. Previous work on learning a gain includes Hoang et al. (1994; 1998); Mallia-Parfitt & Bröcker (2016); Levine & Stuart (2022). However, this previous work involved optimizing a mean-square error matching the filter mean to observations, thus doing state estimation instead of probabilistic estimation. Learning an

ensemble filter analysis step was proposed by McCabe & Brown (2021), but again used a mean-square error loss to match the filter mean to observations.

In our numerical experiments involving an EnKF, we apply automatic differentiation through the filter. Other work that involved automatic differentiation through an EnKF is Chen et al. (2022), albeit for the purpose of learning forecast model parameters instead of parameters of the analysis step.

We also test here the application of our method to learning inflation and localization in an EnKF. There is a large literature on adaptively choosing inflation and localization parameters in EnKFs (e.g., Miyoshi, 2011; Vishny et al., 2024). However, there is no work we are aware of based on picking these parameters to match the true filter.

2. Variational Inference for Parameterized Filters

We recall that the analysis step in (3) is an application of Bayes' Theorem with likelihood $\mathbb{P}(y_{j+1}^\dagger | v_{j+1})$ and prior $\widehat{\Pi}_{j+1}$. Variational inference (Blei et al., 2017) can be used to transform this Bayesian inference problem to the solution of the optimization problem

$$J(q) = D_{\text{KL}}(q \| \widehat{\Pi}_{j+1}) - \mathbb{E}^{v_{j+1} \sim q} [\log \mathbb{P}(y_{j+1}^\dagger | v_{j+1})], \quad (6a)$$

$$\Pi_{j+1} = \arg \min_{q \in \mathcal{P}(\mathbb{R}^d)} J(q), \quad (6b)$$

where D_{KL} is the Kullback–Leibler (KL) divergence.

2.1. Offline and Online Formulations

Recall that the prediction P and analysis $A(\cdot; y_{j+1}^\dagger)$ operators define the filtering update (3). Using the parameterized analysis map A_θ , we define an optimization problem over θ rather than an optimization over densities. We first define a recursion for densities q'_j ,

$$q'_{j+1}(\theta) = A_\theta(Pq'_j(\theta); y_{j+1}^\dagger), \quad q'_0 = \Pi_0,$$

and consider the optimization problem

$$J_{j+1}(\theta) = D_{\text{KL}}(q'_{j+1}(\theta) \| Pq'_j(\theta)) - \mathbb{E}^{v_{j+1} \sim q'_{j+1}(\theta)} [\log \mathbb{P}(y_{j+1}^\dagger | v_{j+1})], \quad (7a)$$

$$J(\theta) = \sum_{j=1}^J J_j(\theta), \quad \theta^* = \arg \min_{\theta} J(\theta). \quad (7b)$$

Note that if there exists a unique θ^* such that $A_{\theta^*} = A$, then this θ^* will be that defined by (7b). Rather than seeking a time-independent A_θ , we can also consider time-dependent

parameters by introducing a new recursion for a density q_j^* ,

$$q_{j+1}^*(\theta) = A_\theta(\mathbb{P}q_j^*(\theta_j^*); y_{j+1}^\dagger), \quad q_0^* = \Pi_0, \quad (8a)$$

$$J_{j+1}(\theta) = D_{\text{KL}}(q_{j+1}^*(\theta) \|\mathbb{P}q_j^*(\theta_j^*)) - \mathbb{E}^{v_{j+1} \sim q_{j+1}^*(\theta)} [\log \mathbb{P}(y_{j+1}^\dagger | v_{j+1})], \quad (8b)$$

$$\theta_{j+1}^* = \arg \min_{\theta} J_{j+1}(\theta). \quad (8c)$$

The recursions (7) and (8) suggest two different methods for finding the filter parameters, which we call *offline* and *online* respectively.

2.2. Implementation

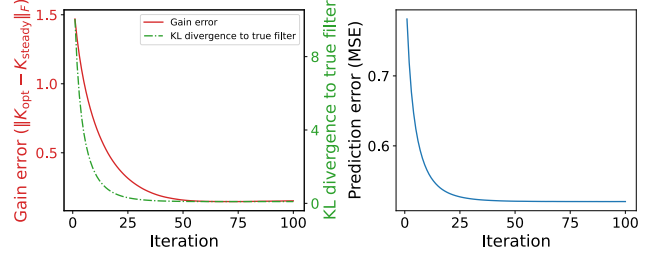
Automatic differentiation is used to optimize both the offline and online cost functions. The expectations are approximated as empirical means using Monte Carlo samples. Some further implementation details of the offline and online methods are as follows:

1. *Offline method.* In the offline method, the sum (7b) is minimized. In so doing, the optimization uses the entire observation set Y_j^\dagger to determine the filter parameters, finding the parameters that achieve the optimal filter “on average” over the time window. These parameters can then be used in a sequential filtering setting. Obtaining the gradient over the entire sum may be expensive from having to differentiate through many compositions of the prediction and analysis steps.
2. *Online method.* In the online method, the cost function (8b) is optimized at each time j , resulting in time-dependent parameters θ_j^* that are optimal for each analysis step. Here, future observations will not affect the optimal parameters at a given time. The gradient only needs to be taken over one analysis step at a time and not the forecast step, reducing the difficulty of automatic differentiation.

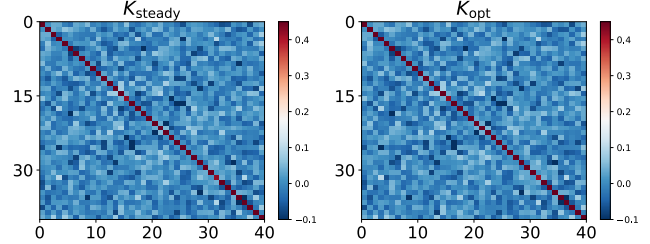
3. Numerical Experiments

We carry out several numerical experiments learning parameters in approximate filters: learning a gain for filtering linear and nonlinear dynamical systems, and learning inflation and localization for an EnKF. Unless otherwise stated, we use the offline method. Appendix A gives the details of the setup for each experiment.

We use `jax` (Bradbury et al., 2018) for automatic differentiation and gradient descent to optimize the cost functions.



(a) Errors as a function of gradient descent iterations. The Frobenius norm of the difference between the learned gain and the steady-state gain (red solid line). The KL divergence between the learned filtering distribution and the Kalman filter distribution (green dashed line). The mean-square prediction error in the filter mean compared to the true trajectory (blue solid line).



(b) A comparison between the steady-state Kalman gain (left) and the learned gain (right).

Figure 1. Gain learning for a linear dynamical system.

3.1. Gain Learning for Linear Dynamical Systems

Here, we assume that $\Psi(\cdot) = A \cdot$ and $h(\cdot) = H \cdot$, and define the following recursion for the mean m_j and covariance C_j

$$\hat{m}_{j+1} = A m_j, \quad \hat{C}_{j+1} = A C_j A^\top + \Sigma \quad (9a)$$

$$m_{j+1} = \hat{m}_{j+1} + K(y_{j+1}^\dagger - H \hat{m}_{j+1}), \quad (9b)$$

$$C_{j+1} = (I - KH) \hat{C}_{j+1} (I - KH)^\top + KTK^\top. \quad (9c)$$

The matrix K , called the *gain*, is unknown, and we will attempt to learn it, i.e., $\theta = K$. We take the filtering distribution at time j to be $\mathcal{N}(m_j(K), C_j(K))$, where $\mathcal{N}(m, C)$ is the multivariate normal distribution with mean m and covariance matrix C . See Appendix B for the justification of the form of this filtering update.

Although for linear dynamics and observations the closed-form solution of the filtering problem is given by the Kalman filter with a time-varying gain K_j , here we test learning a fixed gain K using the offline method. With the exact solution available as a baseline, we can validate our methodology and evaluate its error. We also note that under stability conditions on the system, the Kalman gain K_j will reach a steady-state gain K_{steady} as $j \rightarrow \text{steady}$ (see Appendix C). We test our constant gain against this steady-state gain as well.

We set A to be a randomly generated 40×40 matrix (see Appendix A) and $H = I$. Figure 1a shows the KL diver-

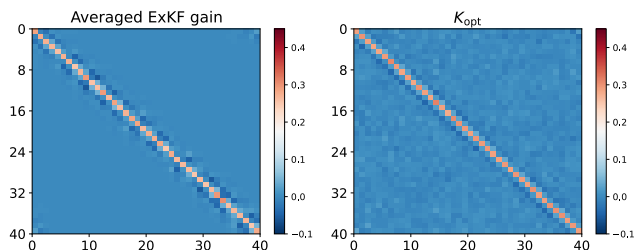


Figure 2. Gains for the Lorenz-96 system. On the left is the extended Kalman filter (ExKF) gain, averaged over 50 iterations. On the right is the learned gain.

gence between the learned filter and the true filter, the norm of the difference between the learned gain and the steady-state gain, and the mean-square prediction error of the state. All of these error measures decrease with more gradient descent iterations for learning K . Figure 1b shows a visual comparison between the steady-state Kalman gain and the learned gain.

We also tested partial observations, wherein only every other state variable is observed (Appendix E.1), as well as the online method (Appendix E.2), with this problem.

3.2. Gain Learning for Nonlinear Dynamical Systems

Here we allow for a nonlinear Ψ (a nonlinear h can easily be accommodated as well). We learn K in the algorithm (9), but replace the prediction step (9a) by

$$\hat{m}_{j+1} = \Psi(m_j), \quad \hat{C}_{j+1} = J_j C_j J_j^\top + \Sigma; \quad (10)$$

here J_j is the Jacobian of Ψ evaluated at m_j . Such a linearization is also made in the extended Kalman filter (Jazwinski, 1970); it also has the form of cycled 3DVar with added covariance propagation (Sanz-Alonso et al., 2023). As in the previous linear problem, we take the filtering distribution at time j to be $\mathcal{N}(m_j(K), C_j(K))$.

We take Ψ to be the 40-dimensional Lorenz-96 model (Lorenz, 1996), a chaotic model of the midlatitude atmosphere often used in data assimilation experiments.

Figure 2 shows a comparison of the averaged extended Kalman filter gain for this system and the one learned by our method. Since we learn a fixed gain and the extended Kalman filter gain is time-varying, and since the extended Kalman filter is only an approximate filter, we do not expect the gains to match. Nonetheless, both methods exhibit gain matrices with a banded and cyclic structure, which is expected due to the correlation structure of states on a circular ring in the Lorenz-96 model. Moreover, due to the symmetry of the model, the optimal fixed gain should be isotropic, which is indeed seen here.

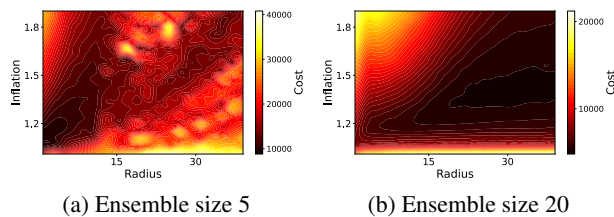


Figure 3. Contours for the cost function as a function of localization radius and inflation.

3.3. Learning Inflation and Localization for EnKF

We test learning inflation and localization for an EnKF (see Appendix D for details). We again use the Lorenz-96 model.

Since the KL divergence cannot be computed with the first argument being an empirical measure, we fit the best Gaussian to the ensemble before computing the KL divergence; this is sometimes referred to as Gaussian projection (Calvello et al., 2022). Furthermore, the covariance matrices are rank deficient whenever the ensemble size $N < d$, which does not allow us to use the formula for the KL divergence for Gaussians. To remedy this rank deficiency we apply Ledoit–Wolf shrinkage (Ledoit & Wolf, 2004) to the covariance matrices, $C \rightarrow (1 - \gamma)C + \gamma I$, and we choose the shrinkage parameter $\gamma = 0.1$.

Figure 3 shows contours of the cost function as a function of inflation and localization radius, for ensemble sizes 5 and 20. In both cases, a larger localization radius generally results in a larger optimal inflation. This is expected because a larger localization radius leads to more sampling error, which can be compensated for by inflation. For both cases the optimal inflation is always greater than 1; this follows from the systematic underestimation of analysis covariance due to sampling error (Sacher & Bartello, 2008).

For ensemble size 5, a localization radius smaller than 5 and inflation between 1.1 and 1.2 results in the lowest cost. With a larger ensemble size, the sampling error in the covariance matrices is lessened, allowing for a less severe localization. Ensemble size 20 thus achieves the lowest cost with a localization radius greater than 20 and inflation between 1.3 and 1.5. Note also that the lowest costs in the contour plot for the 5-member ensemble are higher than those for the 20-member ensemble, indicating that the EnKF gets closer to the true filter with the larger ensemble size.

4. Conclusions

We introduce a framework and methodology for learning parameterized filters and apply it to learning gains for filtering linear and nonlinear dynamical systems, as well as learning inflation and localization for an EnKF.

Future work will investigate the learning of more general filters by approximating the analysis map directly. For instance, θ can be taken to be the weights of a neural network that takes in a forecast ensemble and observations and produces an analysis ensemble. Furthermore, state-dependent gains, such as those that appear in the feedback particle filter (Yang et al., 2013), can be considered.

Here, for the ensemble algorithms, the analysis measures were projected back into the space of Gaussians in order to estimate the KL divergence. Future work will consider sample-based estimators of the KL divergence (Wang et al., 2009) or alternative sample-based objective functions.

References

- Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, April 2017. ISSN 0162-1459. doi: 10.1080/01621459.2017.1285773.
- Boudier, P., Fillion, A., Gratton, S., Gürol, S., and Zhang, S. Data Assimilation Networks. *Journal of Advances in Modeling Earth Systems*, 15(4):e2022MS003353, 2023. ISSN 1942-2466. doi: 10.1029/2022MS003353.
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/google/jax>.
- Calvello, E., Reich, S., and Stuart, A. M. Ensemble Kalman Methods: A Mean Field Perspective. arXiv:2209.11371 (cs, math), September 2022. doi: 10.48550/arXiv.2209.11371.
- Campbell, A., Shi, Y., Rainforth, T., and Doucet, A. Online Variational Filtering and Parameter Learning. In *Advances in Neural Information Processing Systems*, November 2021.
- Carrassi, A., Bocquet, M., Bertino, L., and Evensen, G. Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *Wiley Interdisciplinary Reviews: Climate Change*, 9(5):e535, September 2018. ISSN 1757-7799. doi: 10.1002/wcc.535.
- Carrillo, J. A., Hoffmann, F., Stuart, A. M., and Vaes, U. Statistical Accuracy of Approximate Filtering Methods. arXiv:2402.01593(cs, math), February 2024. doi: 10.48550/arXiv.2402.01593.
- Chen, Y., Sanz-Alonso, D., and Willett, R. Autodifferentiable Ensemble Kalman Filters. *SIAM Journal on Mathematics of Data Science*, 4(2):801–833, June 2022. doi: 10.1137/21M1434477.
- Evensen, G. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994. ISSN 2156-2202. doi: 10.1029/94JC00572.
- Hoang, H., De Mey, P., and Talagrand, O. A simple adaptive algorithm of stochastic approximation type for system parameter and state estimation. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 1, pp. 747–752 vol.1, December 1994. doi: 10.1109/CDC.1994.410863.
- Hoang, S., Baraille, R., Talagrand, O., Carton, X., and De Mey, P. Adaptive filtering: Application to satellite data assimilation in oceanography. *Dynamics of Atmospheres and Oceans*, 27(1):257–281, January 1998. ISSN 0377-0265. doi: 10.1016/S0377-0265(97)00014-6.
- Houtekamer, P. L. and Zhang, F. Review of the Ensemble Kalman Filter for Atmospheric Data Assimilation. *Monthly Weather Review*, 144(12):4489–4532, June 2016. ISSN 0027-0644. doi: 10.1175/MWR-D-15-0440.1.
- Jazwinski, A. H. *Stochastic Processes and Filtering Theory*. Number 64 in Mathematics in Science and Engineering. Academic Press, Inc., New York, 1970.
- Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, March 1960. ISSN 0021-9223. doi: 10.1115/1.3662552.
- Kalnay, E. *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, New York, December 2002. ISBN 978-0-521-79629-3.
- Law, K., Stuart, A., and Zygalkakis, K. *Data Assimilation: A Mathematical Introduction*. Number 62 in Texts in Applied Mathematics. Springer International Publishing, 2015. ISBN 978-3-319-20324-9 978-3-319-20325-6. doi: 10.1007/978-3-319-20325-6.
- Le Gland, F., Monbet, V., and Tran, V.-D. Large sample asymptotics for the ensemble Kalman filter. In *The Oxford Handbook of Nonlinear Filtering*, pp. 598–631. Oxford University Press, 2011. ISBN 978-0-19-953290-2.
- Ledoit, O. and Wolf, M. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, February 2004. ISSN 0047-259X. doi: 10.1016/S0047-259X(03)00096-4.
- Levine, M. and Stuart, A. A framework for machine learning of model error in dynamical systems. *Communications of the American Mathematical Society*, 2(07):283–344, 2022. ISSN 2692-3688. doi: 10.1090/cams/10.

- Lorenz, E. N. Predictability: A problem partly solved. In *Proceedings of a Seminar Held at ECMWF on Predictability, 4-8 September 1995*, volume 1, pp. 1–18, Shinfield Park, Reading, 1996. ECMWF.
- Mallia-Parfitt, N. and Bröcker, J. Assessing the performance of data assimilation algorithms which employ linear error feedback. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(10):103109, October 2016. ISSN 1054-1500. doi: 10.1063/1.4965029.
- Marino, J., Cvitkovic, M., and Yue, Y. A General Method for Amortizing Variational Filtering. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- McCabe, M. and Brown, J. Learning to Assimilate in Chaotic Dynamical Systems. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12237–12250. Curran Associates, Inc., 2021.
- Miyoshi, T. The Gaussian Approach to Adaptive Covariance Inflation and Its Implementation with the Local Ensemble Transform Kalman Filter. *Monthly Weather Review*, 139(5):1519–1535, May 2011. ISSN 0027-0644, 1520-0493. doi: 10.1175/2010MWR3570.1.
- Reich, S. and Cotter, C. *Probabilistic forecasting and Bayesian data assimilation*. Cambridge University Press, 2015.
- Sacher, W. and Bartello, P. Sampling Errors in Ensemble Kalman Filtering. Part I: Theory. *Monthly Weather Review*, 136(8):3035–3049, August 2008. ISSN 0027-0644. doi: 10.1175/2007MWR2323.1.
- Sakov, P. and Bertino, L. Relation between two common localisation methods for the EnKF. *Computational Geosciences*, 15(2):225–237, March 2011. ISSN 1573-1499. doi: 10.1007/s10596-010-9202-6.
- Sanz-Alonso, D., Stuart, A., and Taeb, A. *Inverse Problems and Data Assimilation*. London Mathematical Society Student Texts. Cambridge University Press, Cambridge, 2023. ISBN 978-1-00-941432-6. doi: 10.1017/9781009414319.
- Simon, D. *Optimal State Estimation: Kalman, H_∞ , and Nonlinear Approaches*. John Wiley & Sons, Ltd, 2006. ISBN 978-0-470-04534-3. doi: 10.1002/0470045345.
- Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008.
- Vishny, D. N., Morzfeld, M., Gwartz, K., Bach, E., Dunbar, O., and Hodyss, D. High-dimensional covariance estimation from a small number of samples. *Earth and Space Science Open Archive*, May 2024. doi: 10.22541/essoar.171501094.44068137/v1.
- Wang, Q., Kulkarni, S. R., and Verdu, S. Divergence Estimation for Multidimensional Densities Via k-Nearest Neighbor Distances. *IEEE Transactions on Information Theory*, 55(5):2392–2405, May 2009. ISSN 1557-9654. doi: 10.1109/TIT.2009.2016060.
- Yang, T., Mehta, P. G., and Meyn, S. P. Feedback Particle Filter. *IEEE Transactions on Automatic Control*, 58(10):2465–2480, October 2013. ISSN 1558-2523. doi: 10.1109/TAC.2013.2258825.

A. Experiment Parameters

Linear model experiments: A stable matrix A in the linear forecast dynamics model $\Psi(v) = Av$ was randomly generated as follows. First, a 40×40 random matrix W was generated with i.i.d. entries drawn from a standard Gaussian, i.e., $W_{ij} \sim \mathcal{N}(0, 1)$ for $1 \leq i, j \leq 40$. Then, the matrix was symmetrized by taking the average with its transpose, i.e., $0.5(W + W^\top)$. Finally, the eigenvalues of the matrix were normalized by $\lambda_{\max} + 0.1$, where λ_{\max} is the maximum eigenvalue, and the resulting matrix reconstituted using the change of basis defining the diagonalization.

Experiments were conducted over $J = 1000$ steps. We used $N = 10$ Monte Carlo samples for approximating the expectations. The process noise covariance matrix, Σ , was defined as $\Sigma = QQ^\top + 0.1I$, with Q being a 40×40 matrix with random entries distributed as $\mathcal{N}(0, 0.25)$. The observation matrix and noise covariance matrix, H and Γ , were set to the identity, and the initial state mean, m_0 , to be a vector of ones.

Observations were made at every time step, and the learning rate for gradient descent was set to $\alpha = 10^{-5}$ over 100 iterations.

Nonlinear model experiments: The Lorenz-96 model is given by

$$\frac{dx_i}{dt} = -x_{i-1}(x_{i-2} + x_{i+1}) - x_i + F, \quad (11)$$

where the indices i range from 1 to D and are cyclical. We used $D = 40$ and $F = 8$. The model was integrated with fourth-order Runge–Kutta and a time step of 0.05, and observations assimilated every time step. As before, we use $J = 1000$ time steps, $H = I$, and $\Gamma = I$.

The nonlinear model experiments used a diagonal process noise covariance matrix, $\Sigma = 0.1I$. The learning rate for gradient descent was set to $\alpha = 10^{-5}$ over 100 iterations.

B. Linear 3DVar Filter

We consider a random vector v_j whose law will define the approximate filter:

$$\hat{v}_{j+1} = Av_j + \xi_j, \quad (12a)$$

$$v_{j+1} = \hat{v}_{j+1} + K(y_{j+1}^\dagger + \eta_{j+1} - H\hat{v}_{j+1}), \quad (12b)$$

$$v_0 \sim \mathcal{N}(m_0, C_0), \quad \xi_j \sim \mathcal{N}(0, \Sigma) \text{ i.i.d.}, \quad (12c)$$

$$\eta_{j+1} \sim \mathcal{N}(0, \Gamma) \text{ i.i.d.} \quad (12d)$$

We proceed to show that the corresponding mean and covariance equations are given by (9). Since v_0^\dagger and v_0 are both distributed according to $\mathcal{N}(m_0, C_0)$, $\mathbb{E}[v_0] = \mathbb{E}[v_0^\dagger]$. Then, under the recursions (1) and (12), it follows by induction that $\mathbb{E}[v_j] = \mathbb{E}[v_j^\dagger]$ for all $j \geq 0$. This implies that $\mathbb{E}[y_j^\dagger] = H\mathbb{E}[v_j]$ and that $\mathbb{E}[(y_j^\dagger - H\mathbb{E}[v_j])(y_j^\dagger - H\mathbb{E}[v_j])^\top] = \Gamma$.

The formulae for $\hat{m}_j = \mathbb{E}[\hat{v}_j]$ and $m_j = \mathbb{E}[v_j]$ follow immediately. For the covariance $\hat{C}_j = \mathbb{E}[(\hat{v}_j - \hat{m}_j)(\hat{v}_j - \hat{m}_j)^\top]$ we have

$$\begin{aligned} \hat{C}_{j+1} &= \mathbb{E}[(A(v_j - m_j) + \xi_{j+1})(A(v_j - m_j) + \xi_{j+1})^\top], \\ &= AC_jA^\top + \Sigma. \end{aligned}$$

For $C_j = \mathbb{E}[(v_j - m_j)(v_j - m_j)^\top]$ we have

$$\begin{aligned} C_{j+1} &= \mathbb{E}[(I - KH)(\hat{v}_{j+1} - \hat{m}_{j+1}) + K\eta_{j+1}][(I - KH)(\hat{v}_{j+1} - \hat{m}_{j+1}) + K\eta_{j+1}]^\top, \\ &= (I - KH)\hat{C}_{j+1}(I - KH)^\top + K\Gamma K^\top. \end{aligned}$$

We note that this covariance update equation for C_j coincides with the equation for the error covariance matrix $\mathbb{E}[(v_j^\dagger - v_j)(v_j^\dagger - v_j)^\top]$, and is known as the Joseph formula (Simon, 2006).

C. Kalman Filter

Suppose that $\Psi(\cdot) = A\cdot$ and $h(\cdot) = H\cdot$. Then the solution to the filtering problem is a Gaussian distribution for all time. The update rules for the mean and covariance of the filter are given by the *Kalman filter* (Kalman, 1960): the mean updates are given by

$$\hat{m}_{j+1} = Am_j, \quad (13a)$$

$$m_{j+1} = \hat{m}_{j+1} + K_j(y_{j+1}^\dagger - H\hat{m}_{j+1}), \quad (13b)$$

where the *Kalman gain* K_j is determined by the update rule for the covariances

$$\hat{C}_{j+1} = AC_jA^\top + \Sigma, \quad (14a)$$

$$K_{j+1} = \hat{C}_{j+1}H^\top (H\hat{C}_{j+1}H^\top + \Gamma)^{-1}, \quad (14b)$$

$$C_{j+1} = (I - K_{j+1}H)\hat{C}_{j+1}. \quad (14c)$$

We note that if the covariance is in steady state then $C_{j+1} = C_j = C_{\text{steady}}$ and $\hat{C}_{j+1} = \hat{C}_j = \hat{C}_{\text{steady}}$. The steady-state covariance and gain are given by solving the equations

$$\hat{C}_{\text{steady}} = A(I - K_{\text{steady}}H)\hat{C}_{\text{steady}}A^\top + \Sigma, \quad (15a)$$

$$K_{\text{steady}} = \hat{C}_{\text{steady}}H^\top (H\hat{C}_{\text{steady}}H^\top + \Gamma)^{-1} \quad (15b)$$

for the pair $(\hat{C}_{\text{steady}}, K_{\text{steady}})$, and setting

$$C_{\text{steady}} = (I - K_{\text{steady}}H)\hat{C}_{\text{steady}}. \quad (16)$$

D. Ensemble Kalman Filter

The *ensemble Kalman filter* (Evensen, 1994) resembles the Kalman filter, but estimates means and covariances through Monte Carlo sampling and allows for nonlinear dynamics and observation operators. Here we consider an ensemble square-root filter described in Sakov & Bertino (2011) that holds for a linear observation operator $h(\cdot) = H\cdot$. The update at each step for the ensemble matrix $E_j \in \mathbb{R}^{d \times N}$, whose columns contain the N ensemble members, is given by

$$\hat{E}_{j+1}^{(n)} = \Psi(E_j^{(n)}), \quad n = 1, \dots, N, \quad (17a)$$

$$\hat{m}_{j+1} = \frac{1}{N} \sum_{n=1}^N \hat{E}_{j+1}^{(n)}, \quad (17b)$$

$$\hat{C}_{j+1} = L \circ \left(\frac{\lambda^2}{N-1} (\hat{E}_{j+1} - \hat{m}_{j+1}1^\top)(\hat{E}_{j+1} - \hat{m}_{j+1}1^\top)^\top \right) + \Sigma, \quad (17c)$$

$$K_{j+1} = \hat{C}_{j+1}H^\top (H\hat{C}_{j+1}H^\top + \Gamma)^{-1}, \quad (17d)$$

$$m_{j+1} = \hat{m}_{j+1} + K_{j+1}(y_{j+1}^\dagger - H\hat{m}_{j+1}), \quad (17e)$$

$$E_{j+1} = m_{j+1}1^\top + \lambda(I - KH)^{1/2}(\hat{E}_{j+1} - \hat{m}_{j+1}1^\top), \quad (17f)$$

where $E_j^{(n)}$ is the n th column of the ensemble matrix E_j , λ and L are the inflation factor and localization matrix, respectively (described below); 1 is a vector of ones; and \circ is the Hadamard, or element-wise, product.

The approximate filtering distribution at time j is given by the empirical measure of the ensemble. That is,

$$\Pi_j^{\text{EnKF}} = \frac{1}{N} \sum_{n=1}^N \delta_{E_j^{(n)}}, \quad (18)$$

where δ_x is a Dirac delta function centered at x .

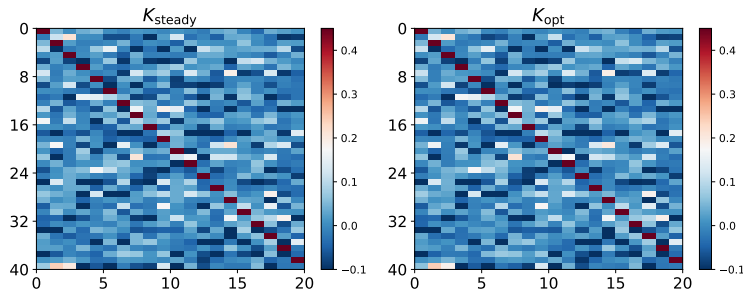


Figure 4. A comparison between the steady-state Kalman gain (left) and the learned gain (right) for the partially observed case.

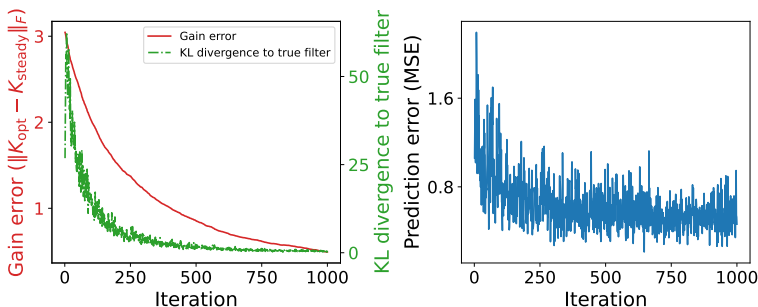


Figure 5. Errors as a function of time step for learning the gain with the online method. Within each time step, 100 gradient descent steps are performed. The Frobenius norm of the difference between the learned gain and the steady-state gain (red solid line). The KL divergence between the learned filtering distribution and the Kalman filter distribution (green dashed line). The mean-square prediction error in the filter mean compared to the true trajectory (blue solid line).

Inflation and localization are both ways of accounting for sampling error in EnKFs. We discuss these briefly but point the reader to the review papers [Houtekamer & Zhang \(2016\)](#); [Carrassi et al. \(2018\)](#) for further details. Sampling error leads to systematic underestimation of covariance matrices in EnKFs ([Sacher & Bartello, 2008](#)). A simple way of accounting for this is by choosing a scalar $\lambda > 1$ and scaling (“inflating”) the ensemble anomalies by it. Localization, based on the idea of spatial decay of correlation, damps elements of the covariance matrix based on their distance in the physical domain. This reduces the impact of spurious correlations. We specifically use the following form of localization matrix:

$$(L)_{ik} = e^{-D_{ik}^2/\ell},$$

where D_{ik} is the distance between the variables i and k and ℓ is the localization length scale. We take the inflation factor λ and localization length scale ℓ as unknown parameters to learn with the method described in this paper.

E. Additional Experiments

The additional experiments follow the linear gain learning example, and use the same parameters as discussed in [Appendix A](#) except for the modifications discussed below.

E.1. Partial observations

For partial observations, the observation matrix H was modified to observe every second state variable. This converged in around 500 gradient descent iterations. [Figure 4](#) shows a faithful recovery of the steady-state gain by this method.

E.2. Online experiments

Here we use the online algorithm discussed in [section 2.1](#) rather than the offline one. In this setting, 100 gradient descent steps were carried out at each time step, for $J = 1000$ time steps. The learning rate was set to $\alpha = 10^{-5}$.

Figure 5 shows the results for learning the gain. Comparing to Figure 1, we can again see convergence towards the steady-state gain. However, the results are not directly comparable since the offline case shows an epoch (pass through the entire dataset) for each iteration, while the online case only involves a single epoch over 1000 time steps.