

Learning optimal filters using variational inference

Eviatar Bach

24 October 2024

University of Reading

*Inverse Problems and Data Assimilation:
A Machine Learning Approach*

Eviatar Bach ^{†‡}, Ricardo Baptista [‡], Daniel Sanz-Alonso [‡], Andrew Stuart [‡]

<https://www.arxiv.org/abs/2410.10523>

I	Inverse Problems	11
1	Bayesian Inversion	13
1.1	Bayesian Inversion	13
1.2	MAP Estimation and Optimization	15
1.3	Well-Posedness of Bayesian Inverse Problems	17
1.4	Model Error	19
1.4.1	Representing Error in Data Space	19
1.4.2	Representing Error in Parameter Space	20
1.4.3	Parameterizing the Forward Model	21
1.5	Bibliography	22
2	Variational Inference	23
2.1	Variational Formulation of Bayes Theorem	23
2.2	Variational Inference	25
2.2.1	Mean-Field Family	25
2.2.2	Gaussian Distributions	27
2.2.3	Mode-Seeking Versus Mean-Seeking Variational Inference	29
2.2.4	Evidence Lower Bound	29
2.3	Bibliography	30
3	Forward Surrogate Modelling	33
3.1	Accelerating Bayesian Inversion	33
3.2	Posterior Approximation Theorem	34
3.3	Accelerating MAP Estimation	36
3.4	Bibliography	37
4	Learning Prior and Regularizers	39
4.1	Learning the Prior	39
4.2	Representing the Prior via a Pushforward	40
4.3	Perturbations to the Prior	41
4.3.1	Smooth Approximation of the Prior	41
4.3.2	Empirical Approximation of the Prior	42

4.4	Learning Regularizers for MAP Estimation	44
4.5	Learning Regularizers for Posterior Approximation	45
4.6	Bibliography	47
5	Transporting to the Posterior	49
5.1	Learning the Prior to Posterior Map	49
5.2	Connection to Variational Inference	51
5.3	Learning Other Posterior Maps	52
5.4	Bibliography	52
6	Learning Dependence on Data	55
6.1	Likelihood-Based Inference	55
6.2	Likelihood-Free Inference	57
6.2.1	Consequences of Block-Triangular Pushforward	58
6.2.2	Learning Block-Triangular Pushforward Maps	62
6.3	Learning Likelihood Models	64
6.4	Amortized MAP Estimation	66
6.5	Bibliography	67
II	Data Assimilation	69
7	Filtering and Smoothing Problems	71
7.1	Formulation of the Filtering Problem	73
7.2	Formulation of the Smoothing Problem	74
7.3	Kalman Filter	75
7.4	3DVar	76
7.5	Extended Kalman Filter (ExKF)	77
7.6	Ensemble Kalman Filter (EnKF)	78
7.6.1	The EnKF Algorithm	78
7.6.2	Inflation	79
7.6.3	Localization	80
7.7	Bootstrap Particle Filter	81
7.8	Optimal Particle Filters	82
7.9	4DVar	83
7.10	Reanalysis	84
7.11	Model Error	85
7.11.1	Model Error: Smoothing	85
7.11.2	Model Error: Filtering	86
7.11.3	Filtering with Small Model Error	86
7.12	Bibliography	88

8	Learning Forecast and Observation Model	91
8.1	The Setting	92
8.2	Expectation Maximization	93
8.2.1	Learning Observation and Model Error Covariances	95
8.2.2	Monte Carlo EM	96
8.3	Auto-Differentiable Kalman Filters	97
8.4	Correcting Model Error Using Analysis Increments	99
8.5	Discussion and Bibliography	100
9	Learning Parameterized Filters and Smoothers	103
9.1	Variational Formulations of Smoothing and Filtering	103
9.1.1	Variational Formulation of Smoothing	104
9.1.2	Variational Formulation of Filtering	104
9.2	Smoothing: State Estimation	107
9.3	Smoothing: Probabilistic Estimation	108
9.3.1	Smoothing: Gaussian Approximate Probabilistic Estimation	108
9.3.2	Smoothing: Amortized Gaussian Approximate Probabilistic Estimation	109
9.4	Filtering: State Estimation	109
9.4.1	3DVar	109
9.4.2	EnKF Gain	112
9.4.3	EnKF Localization and Inflation	114
9.4.4	Optimal Particle Filter	114
9.4.5	Spread-Error Relationship	115
9.4.6	Learning State Estimators in the Presence of Model Error	116
9.5	Filtering: Probabilistic Estimation	117
9.5.1	Learning Filters Using Variational Inference	118
9.5.2	Learning Filters Using Strictly Proper Scoring Rules	119
9.5.3	Bootstrap Particle Filter	120
9.5.4	Optimal Particle Filter	121
9.6	Bibliography	122
10	Learning the Filter or Smoother Using Transport	125
10.1	Learning the Forecast to Analysis Map	125
10.2	Learning Dependence of the Map on Data	127
10.2.1	Minimizing the Energy Distance	128
10.2.2	Composed Maps Learned with Maximum Likelihood	128
10.3	Optimal Particle Filter Transport	130
10.3.1	Learning with Likelihood Weights	131
10.3.2	Learning the Data Dependence	132
10.4	Bibliography	132

11	Data Assimilation Using Learned Forecasting Models	133
11.1	Smoothing and Filtering Using Surrogate Models	133
11.1.1	Smoothing Problem	133
11.1.2	3DVar	135
11.2	Multifidelity Ensemble State Estimation	136
11.2.1	Multi-Model (Ensemble) Kalman Filters	136
11.2.2	Multifidelity Monte Carlo	138
11.2.3	Model Selection and Sample Allocation	141
11.3	Multifidelity Covariance Estimation	141
11.4	Bibliography	143

III Learning Frameworks 145

12	Metrics, Divergences and Scoring Rules	147
12.1	Metrics	147
12.1.1	Metrics on the Space of Probability Measures	147
12.1.2	Total Variation and Hellinger Metrics	148
12.1.3	Transportation Metrics	151
12.1.4	Integral Probability Metrics	153
12.1.5	Maximum Mean Discrepancy and Energy Distance	154
12.1.6	Metrics on the Space of Random Probability Measures	156
12.2	Divergences	157
12.2.1	f-Divergences	157
12.2.2	Relationships between f-Divergences and Metrics	159
12.2.3	Invariance of f-Divergences under Invertible Transformations	161
12.3	Scoring Rules	162
12.3.1	Energy Score	163
12.3.2	Continuous Ranked Probability Score	164
12.3.3	Quantile Score	166
12.3.4	Logarithmic Score	168
12.3.5	Dawid-Sebastiani Score	169
12.3.6	Noise in the Verification	169
12.3.7	Distance-Like Deterministic Scoring Rule	170
12.4	Bibliography	170

13	Unsupervised Learning and Generative Modeling	173
13.1	Density Estimation	174
13.2	Transport Methods	175
13.3	Normalizing Flows	177
13.3.1	Structure in the Optimization Problem for θ	177
13.3.2	Neural ODEs	178
13.4	Score-Based Approaches	180
13.5	Autoencoders	183

13.6 Variational Autoencoders	185
13.7 Generative Adversarial Networks	186
13.8 Bibliography	187
14 Supervised Learning	189
14.1 Neural Networks	190
14.2 Random Features	191
14.3 Gaussian Processes	193
14.3.1 Kernels	193
14.3.2 Regression	194
14.4 Approximation Properties	196
14.5 Bibliography	197
15 Time Series Forecasting	199
15.1 Linear Autoregressive Models	199
15.2 Analog Forecasting	200
15.2.1 Analog Forecasting	200
15.2.2 Kernel Analog Forecasting	201
15.3 Recurrent Structure	202
15.3.1 Markovian Prediction	202
15.3.2 Memory and Prediction	202
15.3.3 Memory and Reservoir Computing	203
15.4 Non-Gaussian Auto-Regressive Models	204
15.5 Bibliography	205
16 Optimization	207
16.1 Gradient Descent	207
16.1.1 Deterministic Gradient Descent	207
16.1.2 Stochastic Gradient Descent	209
16.2 Automatic Differentiation	210
16.2.1 Forward Mode	211
16.2.2 Reverse Mode	212
16.3 Expectation-Maximization	213
16.4 Newton and Gauss-Newton	215
16.4.1 Newton's Method	215
16.4.2 Gauss-Newton	215
16.5 Ensemble Kalman Inversion	216
16.6 Bibliography	217
Bibliography	219
Alphabetical Index	251

E. Luk, E. Bach, R. Baptista, and A. Stuart (2024). *Learning Optimal Filters Using Variational Inference*. arXiv: 2406.18066[cs,math]

Collaborators



Enoch Luk (Caltech)



Ricardo Baptista (Caltech)



Andrew Stuart (Caltech)

Data assimilation

Consider the dynamical system

$$x_{j+1}^\dagger = \Psi(x_j^\dagger) + \xi_j^\dagger, \quad (1a)$$

$$x_0^\dagger \sim \mathcal{N}(m_0, C_0), \quad \xi_j^\dagger \sim \mathcal{N}(0, Q) \text{ i.i.d.} \quad (1b)$$

Data assimilation

Consider the dynamical system

$$x_{j+1}^\dagger = \Psi(x_j^\dagger) + \xi_j^\dagger, \quad (1a)$$

$$x_0^\dagger \sim \mathcal{N}(m_0, C_0), \quad \xi_j^\dagger \sim \mathcal{N}(0, Q) \text{ i.i.d.} \quad (1b)$$

The observations are given by

$$y_{j+1}^\dagger = h(x_{j+1}^\dagger) + \eta_{j+1}^\dagger, \quad (2a)$$

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

Data assimilation

Consider the dynamical system

$$x_{j+1}^\dagger = \Psi(x_j^\dagger) + \xi_j^\dagger, \quad (1a)$$

$$x_0^\dagger \sim \mathcal{N}(m_0, C_0), \quad \xi_j^\dagger \sim \mathcal{N}(0, Q) \text{ i.i.d.} \quad (1b)$$

The observations are given by

$$y_{j+1}^\dagger = h(x_{j+1}^\dagger) + \eta_{j+1}^\dagger, \quad (2a)$$

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

We define, for a fixed integer J ,

$$Y_j^\dagger = \{y_1^\dagger, \dots, y_j^\dagger\}.$$

Data assimilation

Consider the dynamical system

$$x_{j+1}^\dagger = \Psi(x_j^\dagger) + \xi_j^\dagger, \quad (1a)$$

$$x_0^\dagger \sim \mathcal{N}(m_0, C_0), \quad \xi_j^\dagger \sim \mathcal{N}(0, Q) \text{ i.i.d.} \quad (1b)$$

The observations are given by

$$y_{j+1}^\dagger = h(x_{j+1}^\dagger) + \eta_{j+1}^\dagger, \quad (2a)$$

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

We define, for a fixed integer J ,

$$Y_j^\dagger = \{y_1^\dagger, \dots, y_j^\dagger\}.$$

The problem of *filtering* is to obtain $\mathbb{P}(x_j^\dagger | Y_j^\dagger)$ for any $j \leq J$.

Solution to the filtering problem

The solution to the filtering problem has a forecast–assimilation structure: $\hat{\pi}_{j+1} = \mathbb{P}(x_{j+1}^\dagger | Y_j^\dagger)$ and decompose the update $\pi_j \mapsto \pi_{j+1}$ into the following two steps:

Solution to the filtering problem

The solution to the filtering problem has a forecast–assimilation structure: $\hat{\pi}_{j+1} = \mathbb{P}(x_{j+1}^\dagger | Y_j^\dagger)$ and decompose the update $\pi_j \mapsto \pi_{j+1}$ into the following two steps:

$$\begin{aligned} \text{Prediction Step:} \quad & \hat{\pi}_{j+1} = P\pi_j, \\ \text{Analysis Step:} \quad & \pi_{j+1} = A(\hat{\pi}_{j+1}; y_{j+1}^\dagger), \end{aligned} \tag{3}$$

where

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

and

$$A(\pi, y^\dagger)(u) = \frac{\nu_\pi(u, y^\dagger)}{\int_{\mathbb{R}^d} \nu_\pi(u, y^\dagger) du}. \tag{4}$$

Solution to the filtering problem

The solution to the filtering problem has a forecast–assimilation structure: $\hat{\pi}_{j+1} = \mathbb{P}(x_{j+1}^\dagger | y_j^\dagger)$ and decompose the update $\pi_j \mapsto \pi_{j+1}$ into the following two steps:

$$\begin{aligned} \text{Prediction Step:} \quad & \hat{\pi}_{j+1} = P\pi_j, \\ \text{Analysis Step:} \quad & \pi_{j+1} = A(\hat{\pi}_{j+1}; y_{j+1}^\dagger), \end{aligned} \tag{3}$$

where

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

and

$$A(\pi, y^\dagger)(u) = \frac{\nu_\pi(u, y^\dagger)}{\int_{\mathbb{R}^d} \nu_\pi(u, y^\dagger) du}. \tag{4}$$

The analysis step thus multiplies by the likelihood $\mathbb{P}(y_{j+1}^\dagger | x_{j+1})$ and normalizes, corresponding to an application of Bayes' Theorem with the prior $\hat{\pi}_{j+1}$.

Introduction

The aim of probabilistic DA is to estimate A , the analysis step.

Introduction

The aim of probabilistic DA is to estimate A , the analysis step.
Existing filters used in practice are suboptimal and involve numerous tuning parameters.

Introduction

The aim of probabilistic DA is to estimate A , the analysis step.

Existing filters used in practice are suboptimal and involve numerous tuning parameters.

Here we consider learning the analysis step of the DA process to get improved filters:

Introduction

The aim of probabilistic DA is to estimate A , the analysis step.

Existing filters used in practice are suboptimal and involve numerous tuning parameters.

Here we consider learning the analysis step of the DA process to get improved filters:

- Learn tuning parameters in existing filtering methods.

Introduction

The aim of probabilistic DA is to estimate A , the analysis step.

Existing filters used in practice are suboptimal and involve numerous tuning parameters.

Here we consider learning the analysis step of the DA process to get improved filters:

- Learn tuning parameters in existing filtering methods.
- Learn new filtering algorithms closer to optimality.

Motivation

Other work has considered learning analysis steps for the purpose of *state estimation*, trained using RMSE.

Motivation

Other work has considered learning analysis steps for the purpose of *state estimation*, trained using RMSE.

Here, we want to learn a probabilistic filter, so RMSE will not do. We need a cost function that is minimized at, and only at, the filtering distribution.

Variational inference

The *Kullback–Leibler (KL) divergence* measures “distance” between probability distributions q and π :

$$D_{\text{KL}}(q\|\pi) = \int \log\left(\frac{q(u)}{\pi(u)}\right) q(u)du. \quad (5)$$

Variational inference

The *Kullback–Leibler (KL) divergence* measures “distance” between probability distributions q and π :

$$D_{\text{KL}}(q\|\pi) = \int \log\left(\frac{q(u)}{\pi(u)}\right) q(u)du. \quad (5)$$

Fact: $D_{\text{KL}}(q\|\pi) \geq 0$ and is minimized if and only if $q = \pi$.

Variational inference

Theorem (Variational formulation of Bayes' Theorem)

$$J(q) = D_{\text{KL}}(q || \mathbb{P}(u)) - \mathbb{E}^q[\log \mathbb{P}(y|u)], \quad (6a)$$

$$\mathbb{P}(u|y) = \arg \min_{q \in \mathcal{P}} J(q), \quad (6b)$$

where $\mathcal{P}$ is the set of all probability distributions on $\mathbb{R}^d$.

Variational inference

Theorem (Variational formulation of Bayes' Theorem)

$$J(q) = D_{\text{KL}}(q || \mathbb{P}(u)) - \mathbb{E}^q[\log \mathbb{P}(y|u)], \quad (6a)$$

$$\mathbb{P}(u|y) = \arg \min_{q \in \mathcal{P}} J(q), \quad (6b)$$

where $\mathcal{P}$ is the set of all probability distributions on $\mathbb{R}^d$.

Notice that the objective is a tradeoff between closeness to the prior (first term) and closeness to observations (second term).

Variational inference

Theorem (Variational formulation of Bayes' Theorem)

$$J(q) = D_{\text{KL}}(q || \mathbb{P}(u)) - \mathbb{E}^q[\log \mathbb{P}(y|u)], \quad (6a)$$

$$\mathbb{P}(u|y) = \arg \min_{q \in \mathcal{P}} J(q), \quad (6b)$$

where $\mathcal{P}$ is the set of all probability distributions on $\mathbb{R}^d$.

Notice that the objective is a tradeoff between closeness to the prior (first term) and closeness to observations (second term).

With the variational formulation of Bayes' Theorem, we turn the inference problem into an optimization (“variational”) problem.

Mathematical framework

Recall that

$$A(\pi, y^\dagger)(u) = \frac{\nu_\pi(u, y^\dagger)}{\int_{\mathbb{R}^d} \nu_\pi(u, y^\dagger) du}, \quad \pi_{j+1} = A(\hat{\pi}_{j+1}, y_{j+1}^\dagger).$$

Mathematical framework

Recall that

$$A(\pi, y^\dagger)(u) = \frac{\nu_\pi(u, y^\dagger)}{\int_{\mathbb{R}^d} \nu_\pi(u, y^\dagger) du}, \quad \pi_{j+1} = A(\hat{\pi}_{j+1}, y_{j+1}^\dagger).$$

Variational inference can be used to transform the analysis step to an optimization problem

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

$$\pi_{j+1} = \arg \min_q J_{j+1}(q). \quad (7b)$$

Mathematical framework

Recall that

$$A(\pi, y^\dagger)(u) = \frac{\nu_\pi(u, y^\dagger)}{\int_{\mathbb{R}^d} \nu_\pi(u, y^\dagger) du}, \quad \pi_{j+1} = A(\hat{\pi}_{j+1}, y_{j+1}^\dagger).$$

Variational inference can be used to transform the analysis step to an optimization problem

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

$$\pi_{j+1} = \arg \min_q J_{j+1}(q). \quad (7b)$$

Suppose we introduce parameters θ into an approximate analysis map:

$$q_{j+1}(\theta) = A_\theta(Pq_j(\theta); y_{j+1}^\dagger), \quad q_0 = \pi_0, \quad (8)$$

e.g., tuning parameters in the EnKF analysis step or neural network parameters.

Method

We then consider the optimization problem

$$J_{j+1}(\theta) = D_{\text{KL}}(q_{j+1}(\theta) \| P q_j) - \mathbb{E}^{x_{j+1} \sim q_{j+1}(\theta)} [\log \mathbb{P}(y_{j+1}^\dagger | x_{j+1})]. \quad (9)$$

Method

We then consider the optimization problem

$$J_{j+1}(\theta) = D_{\text{KL}}(q_{j+1}(\theta) \| Pq_j) - \mathbb{E}^{x_{j+1} \sim q_{j+1}(\theta)} [\log \mathbb{P}(y_{j+1}^\dagger | x_{j+1})]. \quad (9)$$

Only the θ which recovers the true filtering distribution will minimize the cost function!

Method

We then consider the optimization problem

$$J_{j+1}(\theta) = D_{\text{KL}}(q_{j+1}(\theta) \| Pq_j) - \mathbb{E}^{x_{j+1} \sim q_{j+1}(\theta)} [\log \mathbb{P}(y_{j+1}^\dagger | x_{j+1})]. \quad (9)$$

Only the θ which recovers the true filtering distribution will minimize the cost function!

Then, using automatic differentiation and an approximate filter parameterized by θ , the sum $\sum_{j=1}^J J_j(\theta)$ can be minimized to find the best filter parameters over a trajectory (offline method).

Method

We then consider the optimization problem

$$J_{j+1}(\theta) = D_{\text{KL}}(q_{j+1}(\theta) \| Pq_j) - \mathbb{E}^{x_{j+1} \sim q_{j+1}(\theta)} [\log \mathbb{P}(y_{j+1}^\dagger | x_{j+1})]. \quad (9)$$

Only the θ which recovers the true filtering distribution will minimize the cost function!

Then, using automatic differentiation and an approximate filter parameterized by θ , the sum $\sum_{j=1}^J J_j(\theta)$ can be minimized to find the best filter parameters over a trajectory (offline method).

Alternatively, $J_j(\theta)$ can be minimized at each j (online method).

Method

We then consider the optimization problem

$$J_{j+1}(\theta) = D_{\text{KL}}(q_{j+1}(\theta) \| Pq_j) - \mathbb{E}^{x_{j+1} \sim q_{j+1}(\theta)} [\log \mathbb{P}(y_{j+1}^\dagger | x_{j+1})]. \quad (9)$$

Only the θ which recovers the true filtering distribution will minimize the cost function!

Then, using automatic differentiation and an approximate filter parameterized by θ , the sum $\sum_{j=1}^J J_j(\theta)$ can be minimized to find the best filter parameters over a trajectory (offline method).

Alternatively, $J_j(\theta)$ can be minimized at each j (online method).

jax was used for all the experiments that follow.

Gain learning

We first learn a fixed gain K in a 3DVar-like method:

$$x_{j+1} = \hat{x}_{j+1} + K(y_{j+1}^\dagger + \eta_{j+1} - H\hat{x}_{j+1}), \quad (10)$$

for a linear dynamical system.

Gain learning

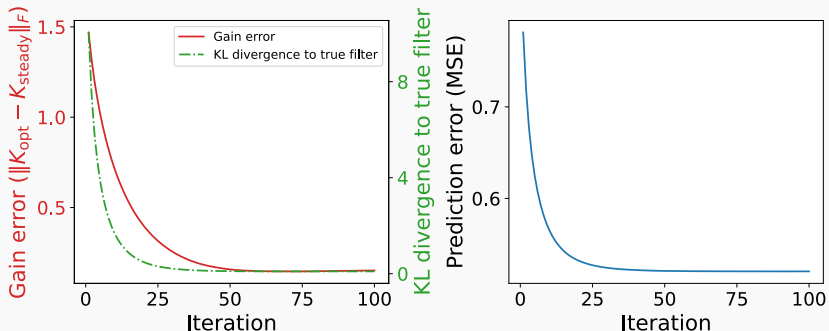
We first learn a fixed gain K in a 3DVar-like method:

$$x_{j+1} = \hat{x}_{j+1} + K(y_{j+1}^\dagger + \eta_{j+1} - H\hat{x}_{j+1}), \quad (10)$$

for a linear dynamical system.

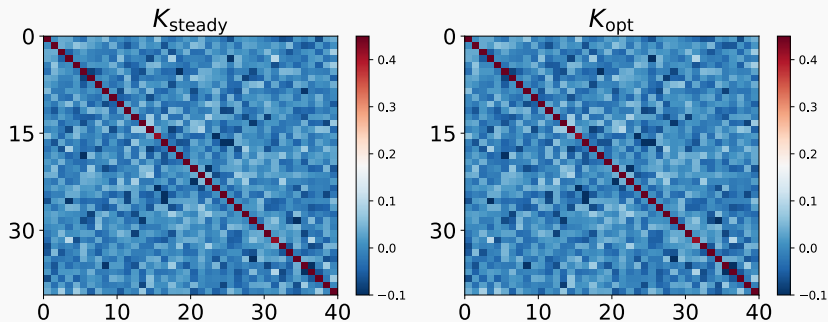
In this case the analytical solution as $J \rightarrow \infty$ is the steady-state Kalman gain.

Gain learning loss



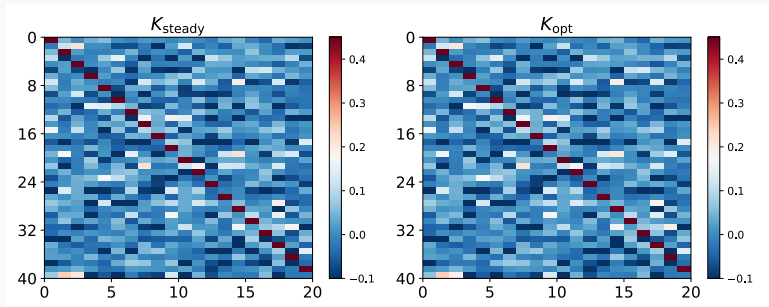
Errors as a function of gradient descent iterations. The Frobenius norm of the difference between the learned gain and the steady-state gain (red solid line). The KL divergence between the learned filtering distribution and the Kalman filter distribution (green dashed line). The mean-square prediction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

Gain learning matrices



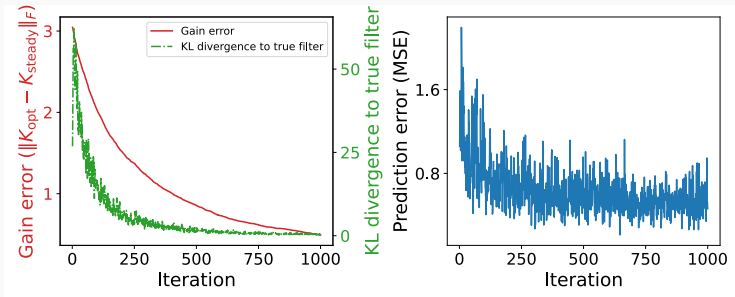
A comparison between the steady-state Kalman gain (left) and the learned gain (right).

50% observations



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

Online learning



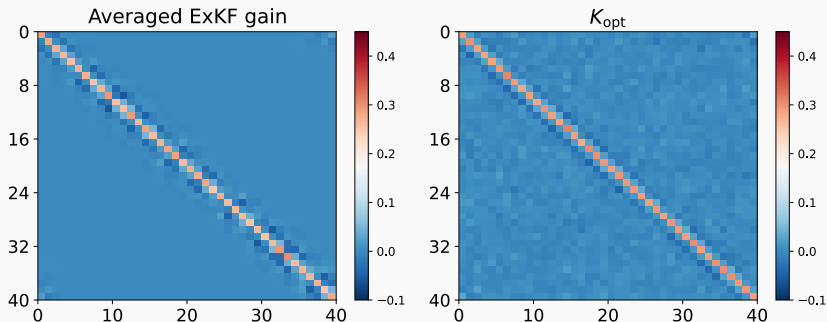
Errors as a function of time step for learning the gain with the online method. The Frobenius norm of the difference between the learned gain and the steady-state gain (red solid line). The KL divergence between the learned filtering distribution and the Kalman filter distribution (green dashed line).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

Gain learning

We now learn a gain matrix for a chaotic nonlinear dynamical system, Lorenz '96.

Gain learning

We now learn a gain matrix for a chaotic nonlinear dynamical system, Lorenz '96.



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.

Learning inflation and localization

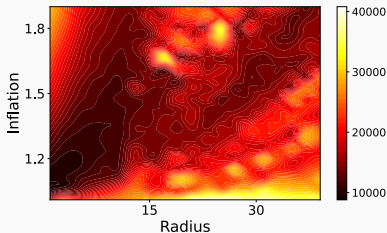
Due to sampling error, empirical covariance matrices need to be inflated and localized to be used in EnKFs, involving tuning parameters.

Learning inflation and localization

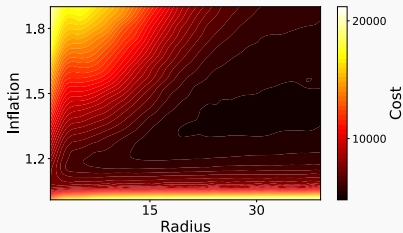
Due to sampling error, empirical covariance matrices need to be inflated and localized to be used in EnKFs, involving tuning parameters.

These parameters are sometimes tuned to minimize RMSE in the EnKF mean. Here, we instead measure distance to the true filter (uncertainty quantification, not only accuracy).

Learning inflation and localization (Lorenz '96)



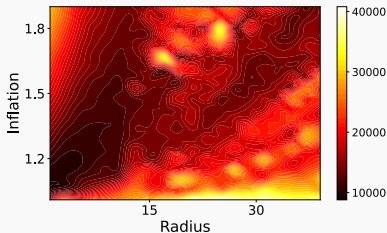
Ensemble size 5



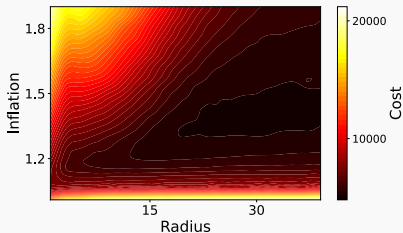
Ensemble size 20

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

Learning inflation and localization (Lorenz '96)



Ensemble size 5



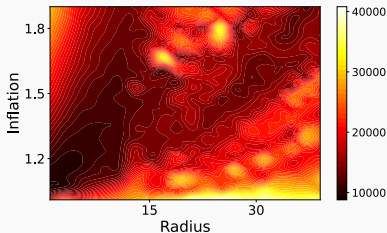
Ensemble size 20

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

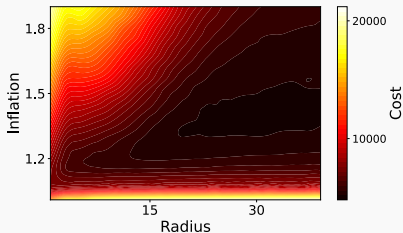
Results:

- Larger localization radius $\rightarrow$ larger optimal inflation.

Learning inflation and localization (Lorenz '96)



Ensemble size 5



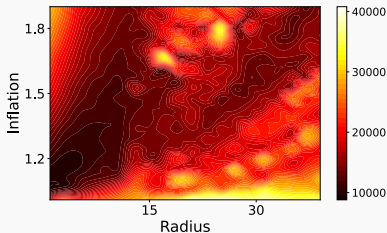
Ensemble size 20

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

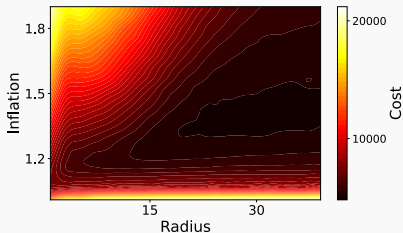
Results:

- Larger localization radius $\rightarrow$ larger optimal inflation.
- Optimal inflation always greater than 1

Learning inflation and localization (Lorenz '96)



Ensemble size 5



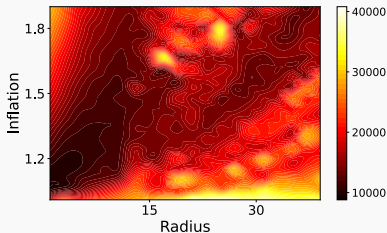
Ensemble size 20

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

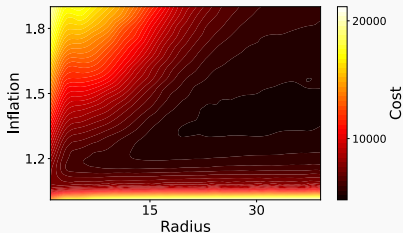
Results:

- Larger localization radius $\rightarrow$ larger optimal inflation.
- Optimal inflation always greater than 1
- Larger ensemble size $\rightarrow$ less severe localization.

Learning inflation and localization (Lorenz '96)



Ensemble size 5



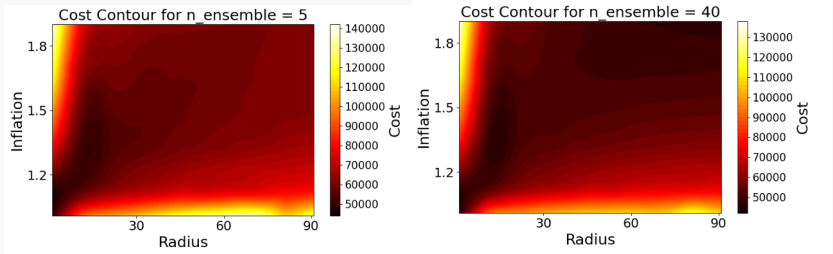
Ensemble size 20

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

Results:

- Larger localization radius $\rightarrow$ larger optimal inflation.
- Optimal inflation always greater than 1
- Larger ensemble size $\rightarrow$ less severe localization.
- EnKF gets closer to the true filter with the larger ensemble size.

Learning inflation and localization (Kuramoto–Sivashinsky)

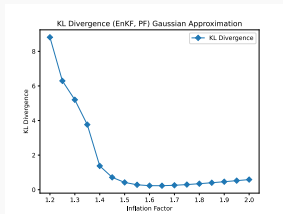


Ensemble size 5

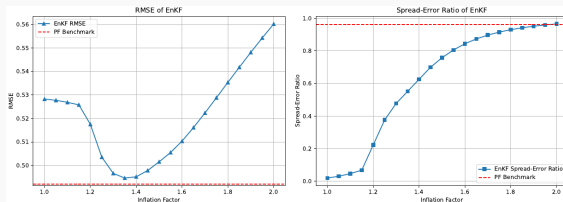
Ensemble size 40

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

Comparison to other cost functions (Lorenz '63)



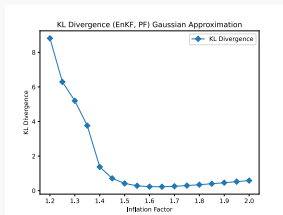
KL to true filter



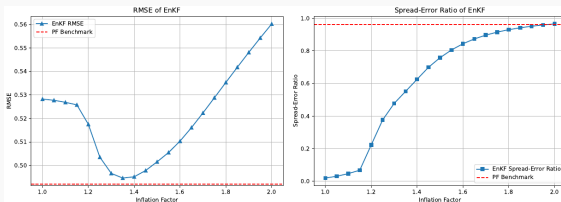
RMSE

Spread-error ratio

Comparison to other cost functions (Lorenz '63)



KL to true filter



RMSE

Spread-error ratio

Variational inference cost function gives different results than RMSE or spread-error!

Conclusions and future work

Variational inference can be used to learn probabilistic filters for DA and set hyperparameters in a Bayesian way.

Conclusions and future work

Variational inference can be used to learn probabilistic filters for DA and set hyperparameters in a Bayesian way.

We show examples applying this method to learning gains for filtering linear and nonlinear dynamical systems, as well as inflation and localization for an EnKF.

Conclusions and future work

Variational inference can be used to learn probabilistic filters for DA and set hyperparameters in a Bayesian way.

We show examples applying this method to learning gains for filtering linear and nonlinear dynamical systems, as well as inflation and localization for an EnKF.

Future work: learning more general analysis steps parameterized by neural networks and scaling to higher-dimensional problems.

Questions?

e.bach@reading.ac.uk

eviatarbach.com

References i



Luk, E., E. Bach, R. Baptista, and A. Stuart (2024). *Learning Optimal Filters Using Variational Inference*. arXiv: 2406.18066[cs,math].