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

Eviatar Bach[♯♮], Ricardo Baptista[♯], Daniel Sanz-Alonso[♭], Andrew Stuart[♯]

[♯] Caltech
[♮] University of Reading
[♭] University of Chicago

# Introduction

## Aim and Overview

The aim of the notes is to demonstrate the potential for ideas in machine learning to impact on the fields of inverse problems and data assimilation. The perspective is one that is primarily aimed at researchers from inverse problems and/or data assimilation who wish to see a mathematical presentation of machine learning as it pertains to their fields. As a by-product of the presentation we present a succinct mathematical treatment of various topics in machine learning. The material on machine learning, along with some other related topics, is summarized in Part III, Appendix. Part I of the notes is concerned with inverse problems, employing material from Part III; Part II of the notes is concerned with data assimilation, employing material from Parts I and III.

## Existing Review Articles

This is a rapidly evolving research area and there are already several articles that review aspects of the material we will cover. See, for example, [14] for uses of ML in inverse problems, and [32, 60] and Chapter 10 of [54] for uses of ML in data assimilation.

## Prerequisites

Even though the subjects of inverse problems and data assimilation are succinctly reviewed in these notes, there remains an assumption of previous knowledge of these topics. The book [278] (also available in closely related form on arXiv) provides a good background on these topics, and we have tried to maintain similar notation. These notes also assume familiarity with linear algebra, probability, multivariable calculus, and matrix calculus.

## Notation

Throughout the notes we adopt the following notational conventions:

- **Sets.** $\mathbb{N}$ denotes the positive integers $\{1, 2, 3, \cdots\}$, and $\mathbb{Z}^+ := \mathbb{N} \cup \{0\} = \{0, 1, 2, 3, \cdots\}$ denotes the non-negative integers. $\mathbb{R}^d$ denotes the set of $d$-dimensional real vectors, and $\mathbb{R}^+$ denotes the non-negative reals. The set $B(u, \delta) \subset \mathbb{R}^d$ denotes the open ball of radius $\delta$ at $u$ in $\mathbb{R}^d$, in the Euclidean norm.
- **Linear Algebra** The symbol $I_d \in \mathbb{R}^{d\times d}$ denotes the identity matrix on $\mathbb{R}^d$, and Id denotes the identity mapping. We denote the Euclidean norm on $\mathbb{R}^d$ by $|\cdot|$, noting that it is induced by the inner-product $\langle a, b\rangle = a^\top b$. We also use $|\cdot|$ to denote the induced norm on matrices and $|\cdot|_F$ to denote the Frobenius norm.

  We say that symmetric matrix $A$ is positive definite (resp. positive semi-definite) if $\langle u, Au\rangle$ is positive (resp. non-negative) for all $u \neq 0$, sometimes denoting this by $A > 0$ (resp. $A \geq 0$). We let $\mathbb{R}^{d\times d}_{\text{sym}}$ denote the subset of symmetric matrices in $\mathbb{R}^{d\times d}$ and $\mathbb{R}^{d\times d}_{\text{sym},>}$ the subset of positive definite symmetric matrices and $\mathbb{R}^{d\times d}$ and $\mathbb{R}^{d\times d}_{\text{sym},\geq}$ the subset of positive semidefinite symmetric matrices. We often wish to use covariance-weighted inner-product and norm and, to this end, for covariance matrix $A > 0$, we define $|v|_A^2 = v^\top A^{-1} v$. This norm $|\cdot|_A$ is induced by the weighted Euclidean inner-product $\langle \cdot\,,\ \cdot\rangle_A := \langle \cdot\,, A^{-1}\cdot\rangle$. The outer product $\otimes$ between two vectors $a, b \in \mathbb{R}^d$ is defined by the following identity, assumed to hold for all vectors $c \in \mathbb{R}^d : (a \otimes b)c = \langle b, c\rangle a$. We also use $\otimes$ to denote the Kronecker product between matrices. We use det and Tr to denote the determinant and trace functions on matrices. We use $\mathrm{vec}(\cdot)$ to denote the vectorization operation on matrices.

- **Probability** To simplify we mostly consider probability measures which have a probability density function (with respect to Lebesgue measure). For this reason we will blur the distinction between probability measures and probability density functions. In particular, when a random variable $u$ has probability density function $\rho$ we will write $u \sim \rho$. Occasionally we will need to employ Dirac masses. In this context we will use the notational convention that the Dirac mass at point $v$ has "density" $\delta(\cdot - v)$, also denoted by $\delta_v(\cdot)$.

  Given probability density function $\rho$, defined on $\mathbb{R}^{d_z}$, and function $g : \mathbb{R}^{d_z} \to \mathbb{R}^d$, $\rho_g$ denotes the probability density function of the random variable $g(z)$ where $z \sim \rho$. We refer to $\rho_g$ as the *pushforward* of $\rho$ under $g$ and write $\rho_g$ as $g_\sharp\rho$. Formally the pushforward notation $\rho_g = g_\sharp\rho$ means that $\mathbb{P}^{\rho_g}(A) = \mathbb{P}^{\rho}(g^{-1}(A))$ for any Borel set $A$. We note that, in some other texts, the pushforward is denoted by $g\#\rho = \rho_g$ or $g_\star\rho = \rho_g$.

  We denote by $\mathbb{P}(\cdot), \mathbb{P}(\cdot \mid \cdot)$ the probability density function of a random variable and its conditional probability density function, respectively. When random variables $u$ and $v$ are independent we write $u \perp\!\!\!\perp v$. For jointly varying random variables $(u, v)$ we let $u|v$ denote the distribution of random variable $u$, given a specific realization of random variable $v$.

  Given $f : \mathbb{R}^d \mapsto \mathbb{R}$ we denote by

$$\mathbb{E}^{\rho}[f] = \int_{\mathbb{R}^d} f(u)\rho(u)\,du$$

  the expectation of $f$ with respect to probability density function $\rho$ on $\mathbb{R}^d$. On occasion, we also denote this expectation by $\rho(f)$. We let $\mathcal{P}(\mathbb{R}^d)$ denote the space of all probability measures over $\mathbb{R}^d$; we simply write $\mathcal{P}$ if the Euclidean space $\mathbb{R}^d$ is clear.

- **Calculus** Given function $f : \mathbb{R}^d \to \mathbb{R}$ we denote by $Df : \mathbb{R}^d \to \mathbb{R}^d$ the gradient and by $D^2 f : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ the Hessian. For more general functions $f : \mathcal{V} \to \mathbb{R}$ acting on elements $v$ of vector space $\mathcal{V}$ we write $Df$ to denote the derivative. We also directly consider functions $f : \mathbb{R}^d \to \mathbb{R}^d$ with Jacobian matrix $Df : \mathbb{R}^d \to \mathbb{R}^{d\times d}$. When there are two potentially varying arguments and we wish to indicate differentiation with respect to only one of them, say $v$, we will indicate this with a subscript: $D_v$ (and similarly for second derivatives). When we contract a derivative to obtain divergence we write div : for example, for $f : \mathbb{R}^d \to \mathbb{R}$, $\mathrm{Tr}\, Df = \mathrm{div}\, f$. We use $f \circ g$ to indicate the composition of $f$ and $g$, assuming input dimensions (of $f$) and output dimensions (of $g$) are compatible, allowing composition.

## Acknowledgments

EB gratefully acknowledges the support from Caltech's Foster and Coco Stanback Postdoctoral Fellowship, the Schmidt Foundation, and the support through federal funding provided to AMS. RB also gratefully acknowledges support for the von Kármán Postdoctoral Instructorship from Caltech and AFOSR MURI programs. DSA is grateful for the support of DOE and NSF (US) and the BBVA Foundation (Spain). AMS would like to acknowledge the various US Government Agencies that have generously supported research in areas related to these notes: AFOSR, ARO, DARPA, DoD, NSF and ONR.

All the authors thank the students who took ACM270 at Caltech in Spring 2024. The lecture notes for this course constituted an early draft of these notes. The students' comments and questions have helped to improve them considerably. The authors are also grateful to Bohan Chen and Jochen Bröcker.

**Warning and Request** This is a first draft of a set of notes that the authors aim to publish on arXiv in the future. As an early draft the notes are likely to contain numerous mathematical errors, incomplete bibliographical information, inconsistencies in notation, and typographical errors. The authors would be extremely grateful for all feedback that might help eliminate any of these issues.

# Contents

# Part I

# Inverse Problems

# Chapter 1

## Bayesian Inversion

A forward model $G : \mathbb{R}^d \to \mathbb{R}^k$ specifies output $y \in \mathbb{R}^k$ from input $u \in \mathbb{R}^d$ by the relationship $y = G(u)$. The related inverse problem is to recover unknown parameter $u \in \mathbb{R}^d$ from data $y \in \mathbb{R}^k$ defined by

$$y = G(u) + \eta, \tag{1.1}$$

where $\eta \in \mathbb{R}^k$ denotes observation noise. In the Bayesian approach to the inverse problem we view $(u, y)$ as jointly varying random variables and the solution of the inverse problem is the posterior distribution on parameter $u$ given a specific instance of the data $y$. In Section 1.1 we formulate Bayesian inverse problems and state Bayes Theorem, giving an explicit expression for the posterior probability density function. In Section 1.2 we connect Bayesian inversion to the classical optimization based approach to inverse problems. Section 1.3 describes the well-posedness of the Bayesian formulation, using the Hellinger distance from Chapter 12. In Section 1.4 we discuss various perspectives on model error. Section 1.5 contains bibliographical remarks.

## 1.1 Bayesian Inversion

We view $(u, y) \in \mathbb{R}^d \times \mathbb{R}^k$ as a random variable, whose distribution is specified by means of the identity (1.1) and the following assumption on the distribution of $(u, \eta) \in \mathbb{R}^d \times \mathbb{R}^k$ :

**Assumption 1.1.** *The random variable $(u, \eta) \in \mathbb{R}^d \times \mathbb{R}^k$ is defined by assuming that $u \perp\!\!\!\perp \eta$, that $u \sim \rho(u)$ and that $\eta \sim \nu(\eta)$.*

Combining this assumption with the relationship between $u$, $y$ and $\eta$ postulated in equation (1.1) we can define the Bayesian inverse problem, a specific and important subclass of general Bayesian inference. We refer to $\rho(u)$, the probability density function of $u$, as the *prior* probability density function. Given identity (1.1), then for fixed $u \in \mathbb{R}^d$, the distribution of $y$ given $u$ defines the *likelihood*:

$$y|u \sim \mathsf{l}(y|u) := \nu\big(y - G(u)\big). \tag{1.2}$$

The posterior probability density function is the conditional distribution of $u$ given $y$, that is the distribution of random variable $u|y$, and is the solution to the Bayesian formulation of the inverse problem.

The primary computational challenge associated with Bayesian inversion is that it is an infinite dimensional problem. In particular it is important to appreciate that, although Bayes theorem (which follows) delivers a formula for the probability density function of $u|y$, the task of obtaining information from this probability density function, for example by drawing many samples from it, is, in general, a substantial one.

**Theorem 1.2.** *Let Assumption 1.1 hold, assume that $u, y$ and $\eta$ satisfy (1.1), and that*

$$Z = Z(y) := \int_{\mathbb{R}^d} \nu(y - G(u))\rho(u)\, du > 0.$$

*Then $u|y \sim \pi^y(u)$, where*[1]

$$\pi^y(u) = \frac{1}{Z}\nu(y - G(u))\rho(u). \tag{1.3}$$

*Proof.* We denote by $\mathbb{P}(\cdot)$ the probability density function of a random variable; and we denote by $\mathbb{P}(\cdot|\cdot)$ the probability density function conditional on the second argument. The standard laws of conditional probability give the two identities

$$\begin{aligned}\mathbb{P}(u, y) &= \mathbb{P}(u|y)\,\mathbb{P}(y), \text{ if } \mathbb{P}(y) > 0,\\ \mathbb{P}(u, y) &= \mathbb{P}(y|u)\,\mathbb{P}(u), \text{ if } \mathbb{P}(u) > 0.\end{aligned}$$

The marginal probability density function on $y$ is given by

$$\begin{aligned}\mathbb{P}(y) &= \int_{\mathbb{R}^d} \mathbb{P}(u, y)\, du\\ &= \int_{\mathbb{R}^d} \mathbb{P}(y|u)\,\mathbb{P}(u)\, du = Z\end{aligned}$$

and so the assumption that $Z > 0$ ensures that the data could indeed have been obtained from the posulated model. Since $Z > 0$ we have, by combining the two identities above, the desired result

$$\mathbb{P}(u|y) = \frac{1}{\mathbb{P}(y)}\,\mathbb{P}(y|u)\,\mathbb{P}(u) = \frac{1}{\mathbb{P}(y)}\nu(y - G(u))\rho(u). \tag{1.4}$$

□

Remark 1.3. We write the density of the joint distribution of $(y, u) \in \mathbb{R}^k \times \mathbb{R}^d$ as $\gamma(y, u)$. We write the marginal density on $y \in \mathbb{R}^k$ as $\kappa(y)$. Note that $\kappa(y) = Z$. ♢

Remark 1.4. When $G$ is linear and $(u, \eta)$ is Gaussian then the posterior on $u|y$ is Gaussian. This setting provides numerous useful explicitly solvable inverse problems in which the posterior is characterized by its mean and covariance; this reduces the infinite dimensional problem to a finite dimensional problem for a vector (mean) in $\mathbb{R}^d$ and a symmetric, positive matrix (covariance) in $\mathbb{R}^{d\times d}$. ♢

Remark 1.5. We have concentrated on formulating Bayes Theorem around equation (1.1) in which the noise appears additively. However it is not restricted to this setting. Section 13.1 provides an explicit example, going beyond this additive noise setting, to formulate density estimation in a Bayesian fashion. ♢

[1] When there is no possibility of confusion, we will simply write $\pi(u)$ for the posterior probability density function, rather than $\pi^y(u)$.

## 1.2 MAP Estimation and Optimization

The infinite dimensional posterior distribution $\pi^y(u)$ contains all knowledge about the parameter $u$, given Assumptions 1.1 and equation (1.1). It is often useful, however, to extract finite dimensional information from the posterior distribution which summarizes it. Remark 1.4 identifies a specific situation where this finite dimensional summary, along with the fact that the posterior is Gaussian, fully characterize the posterior. In general, however, we must seek finite dimensional summaries that may not fully characterize the posterior. One natural summary is the posterior mode or MAP estimator:

**Definition 1.6.** A *maximum a posteriori (MAP) estimator* of $u$ given data $y$ is defined as any point $u_{\text{MAP}}$ satisfying

$$u_{\text{MAP}} \in \arg\max_{u \in \mathbb{R}^d} \pi^y(u).$$

◊

We now show how MAP estimation connects with classical optimization based approaches to inversion. Recall that the posterior probability density function $\pi^y(u)$ on $u|y$ from Theorem 1.2 has the form

$$\pi^y(u) = \frac{1}{Z}\nu(y - G(u))\rho(u).$$

We define a *loss function*

$$\mathsf{L}(u) = -\log \nu(y - G(u)) = -\log \mathsf{l}(y|u), \tag{1.5}$$

and a *regularizer*

$$\mathsf{R}(u) = -\log \rho(u). \tag{1.6}$$

Adding the loss function and regularizer we obtain an *objective function* of the form

$$\mathsf{J}(u) = \mathsf{L}(u) + \mathsf{R}(u). \tag{1.7}$$

Furthermore

$$\pi^y(u) = \frac{1}{Z}\nu(y - G(u))\rho(u) \propto e^{-\mathsf{J}(u)}.$$

Thus the MAP estimator can be rewritten in terms of $\mathsf{J}$ as follows:

$$\begin{aligned} u_{\text{MAP}} &\in \arg\max_{u \in \mathbb{R}^d} \pi^y(u) \\ &= \arg\min_{u \in \mathbb{R}^d} \mathsf{J}(u); \end{aligned}$$

minimizing the objective function $\mathsf{J}(\cdot)$ is equivalent to determining a MAP estimator by maximizing the posterior probability density function $\pi^y(\cdot)$.

Remark 1.7. Setting the regularizer to zero, known as choosing a *flat prior*, we obtain the maximum likelihood estimation (MLE) problem. ◊

**Example 1.8** (Gaussian Observational Noise and Gaussian Prior). If $\eta = \mathcal{N}(0, \Gamma)$, then

$$\nu(y - G(u)) \propto \exp(-\frac{1}{2}|y - G(u)|_\Gamma^2).$$

Thus in this case the loss is

$$\mathsf{L}(u) = \frac{1}{2}|y - G(u)|_\Gamma^2,$$

a $\Gamma$-weighted $\ell_2$ loss.

Now assume that the prior is a centered Gaussian $\rho(u) = \mathcal{N}(0, \widehat{C})$, where $\widehat{C}$ is positive. Ignoring $u$-independent normalization factors, which appear as constant shifts in $\mathsf{J}(\cdot)$ and therefore can be ignored from the viewpoint of minimization, we may take the regularizer to be

$$\mathsf{R}(u) = \frac{1}{2}|u|_{\widehat{C}}^2.$$

Combining these assumptions we obtain a canonical objective function

$$\mathsf{J}(u) = \frac{1}{2}|y - G(u)|_\Gamma^2 + \frac{1}{2}|u|_{\widehat{C}}^2. \tag{1.8}$$

We refer to minimization of (1.8) as the *Tikhonov-Phillips regularized inverse problem*. If $\Gamma \propto \mathrm{Id}$ and $\widehat{C} = \lambda^{-1}\Gamma$ then this reduces to the classical *Tikhonov regularized inverse problem* with objective function

$$\mathsf{J}(u) = \frac{1}{2}|y - G(u)|^2 + \frac{\lambda}{2}|u|^2. \tag{1.9}$$

Finally we note that if $G(u) = Lu$ for matrix $L \in \mathbb{R}^{k \times d}$ then the MAP estimator related to the Tikhonov-Phillips regularized inverse problem is unique and given by point $m$ solving

$$C^{-1} = \widehat{C}^{-1} + L^\top \Gamma^{-1} L, \tag{1.10a}$$

$$C^{-1} m = L^\top \Gamma^{-1} y. \tag{1.10b}$$

In this case, as discussed in Remark 1.4, the posterior is Gaussian, and in fact $\pi^y(u) = \mathcal{N}(m, C)$. This fact may be seen by completing the square in the formula for the posterior. We note that the posterior mean, in this linear Gaussian setting, is equal to the (in this case unique) MAP estimator. ♢

**Example 1.9** ($\ell_1$ Regularizer – Laplace Prior). Now consider the setting where

$$\rho(u) \propto \exp\Big(-\lambda \sum_{i=1}^{d} |u_i|\Big) = \exp(-\lambda|u|_1).$$

This is known as a Laplace distribution. Then $\mathsf{R}(u) = \lambda|u|_1$, an $\ell_1$ regularizer. Combining this prior with the weighted $\ell_2$ loss above, we obtain the objective function

$$\mathsf{J}(u) = \frac{1}{2}|y - G(u)|_\Gamma^2 + \lambda|u|_1.$$

It is well known that regularizers of this type promote sparse solutions when $\mathsf{J}(\cdot)$ is minimized. However it is important to appreciate that samples from the underlying posterior distribution are typically not sparse. ♢

## 1.3 Well-Posedness of Bayesian Inverse Problems

The MAP estimator is an unstable quantity in the following sense: it may change discontinuously with respect to changes in the problem specification, such as the data $y$. This is because the minimizer of an objective function is not, in general, continuous with respect to changes in the objective. Such instability is a general feature of all optimization based approaches to inversion, unless there is some convexity built in via the specifics of a particular problem. On the other hand, the Bayesian formulation of the inverse problem leads to stability in broad generality: small changes in the problem specification lead to small changes in the posterior density. Here we prove a prototypical result of this type.

We consider two different likelihoods

$$\mathsf{l}(y|u) = \nu(y - G(u)) \quad \text{and} \quad \mathsf{l}_\delta(y|u) = \nu(y - G_\delta(u))$$

associated with two different forward models $G(u)$ and $G_\delta(u)$. Assuming the associated Bayesian inverse problems both adopt the same prior $\rho$ we obtain two different posteriors of the form

$$\pi^y(u) = \frac{1}{Z}\mathsf{l}(y|u)\rho(u) \quad \text{and} \quad \pi^y_\delta(u) = \frac{1}{Z_\delta}\mathsf{l}_\delta(y|u)\rho(u),$$

where $Z, Z_\delta$ are the corresponding normalizing constants. Our aim is to show that, if $\mathsf{l}(y|u)$ and $\mathsf{l}_\delta(y|u)$ are close then so are the associated posteriors. To this end we make the following assumptions about the likelihoods, in which we view the data $y$ as fixed.

**Assumption 1.10.** *The data defined by* (1.1) *and Assumption 1.10 has positive probability under the resulting joint distribution on* $(u, y)$*, so that* $Z > 0$*. There exist* $\delta^+ > 0$ *and* $K_1, K_2 < \infty$ *such that, for all* $\delta \in (0, \delta^+)$,

- *(i)* $|\sqrt{\mathsf{l}(y|u)} - \sqrt{\mathsf{l}_\delta(y|u)}| \le \varphi(u)\delta$*, for some* $\varphi(u)$ *satsifying* $\mathbb{E}^\rho[\varphi^2(u)] \le K_1^2$*;*
- *(ii)* $\sup_{u\in\mathbb{R}^d}(|\sqrt{\mathsf{l}(y|u)}| + |\sqrt{\mathsf{l}_\delta(y|u)}|) \le K_2$.

Recall the Hellinger distance $\mathsf{D}_{\mathrm{H}}(\cdot,\cdot)$ from Chapter 12. The main result of this section is:

**Theorem 1.11.** *Under Assumptions 1.1 and 1.10, there exists* $c \in (0, +\infty)$ *and* $\Delta > 0$ *such that, for all* $\delta \in (0, \Delta)$,

$$\mathsf{D}_{\mathrm{H}}(\pi^y, \pi^y_\delta) \le c\delta.$$

Remark 1.12. We make two observations about Theorem 1.11:

- After using Lemma 12.4, the theorem ensures that expectations of functions of $u$, with $u$ distributed according to $\pi^y$ and $\pi^y_\delta$ respectively, are order $\delta$ apart, provided sufficient moments of those functions are available;
- The theorem is prototypical of a variety of stability results for Bayesian inversion. Here we have viewed data $y$ as fixed and considered error in the likelihood arising from approximation of the forward model; in Chapter 3 we establish a similar result in the context of machine learning approximations to the Bayesian inverse

problem where the forward model is replaced by a surrogate. It is also possible to estimate changes in the posterior, in the Hellinger metric, with respect to changes in the data (which of course changes the likelihood).

$\diamondsuit$

*Proof of Theorem 1.11.* To prove Theorem 1.11, we first characterize the stability of the normalization constants. We show that, under Assumption 1.10, there exist $\Delta > 0$ and $c_1, c_2 \in (0, +\infty)$ such that

$$|Z - Z_\delta| \le c_1 \delta \quad \text{and} \quad Z, Z_\delta > c_2, \quad \text{for } \delta \in (0, \Delta).$$

To see this, noting that $Z = \int \mathsf{l}(y|u)\rho(u)\,du$ and $Z_\delta = \int \mathsf{l}_\delta(y|u)\rho(u)\,du$, we have

$$\begin{aligned}
|Z - Z_\delta| =& \Big| \int \big(\mathsf{l}(y|u) - \mathsf{l}_\delta(y|u)\big)\rho(u)\,du \Big| \\
&\le \Big( \int \Big| \sqrt{\mathsf{l}(y|u)} - \sqrt{\mathsf{l}_\delta(y|u)} \Big|^2 \rho(u)\,du \Big)^{1/2} \Big( \int \Big| \sqrt{\mathsf{l}(y|u)} + \sqrt{\mathsf{l}_\delta(y|u)} \Big|^2 \rho(u)\,du \Big)^{1/2} \\
&\le \Big( \int \delta^2 \varphi(u)^2 \rho(u)\,du \Big)^{1/2} \Big( \int K_2^2 \rho(u)\,du \Big)^{1/2} \\
&\le K_1 K_2 \delta, \quad \delta \in (0, \delta^+).
\end{aligned}$$

Therefore, for $\delta \le \Delta := \min\{\frac{Z}{2K_1K_2}, \delta^+\}$, we have

$$Z_\delta \ge Z - |Z - Z_\delta| \ge \frac{1}{2} Z.$$

Since $Z$ is assumed positive we deduce the lower bound on $Z_\delta$. and take $c_2 = \frac{1}{2}Z$. The Lipschitz stability result follows follows by taking $c_1 = K_1 K_2$.

We now study the Hellinger distance between the two posteriors, aiming to use the preceding estimate on the stability of the normalization constants. To this end we break the total error into two contributions, one reflecting the difference between $Z$ and $Z_\delta$, and the other the difference between $\mathsf{l}$ and $\mathsf{l}_\delta$:

$$\begin{aligned}
\mathsf{D}_{\mathrm{H}}(\pi^y, \pi^y_\delta) &= \frac{1}{\sqrt{2}} \Big\| \sqrt{\pi^y} - \sqrt{\pi^y_\delta} \Big\|_{L^2} \\
&= \frac{1}{\sqrt{2}} \Big\| \sqrt{\frac{\mathsf{l}\rho}{Z}} - \sqrt{\frac{\mathsf{l}\rho}{Z_\delta}} + \sqrt{\frac{\mathsf{l}\rho}{Z_\delta}} - \sqrt{\frac{\mathsf{l}_\delta\rho}{Z_\delta}} \Big\|_{L^2} \\
&\le \frac{1}{\sqrt{2}} \Big\| \sqrt{\frac{\mathsf{l}\rho}{Z}} - \sqrt{\frac{\mathsf{l}\rho}{Z_\delta}} \Big\|_{L^2} + \frac{1}{\sqrt{2}} \Big\| \sqrt{\frac{\mathsf{l}\rho}{Z_\delta}} - \sqrt{\frac{\mathsf{l}_\delta\rho}{Z_\delta}} \Big\|_{L^2}.
\end{aligned}$$

From the stability estimate on the normalization constants we have, for $\delta \in (0, \Delta)$,

$$\begin{aligned}\left\|\sqrt{\frac{\mathsf{l}\rho}{Z}} - \sqrt{\frac{\mathsf{l}\rho}{Z_\delta}}\right\|_{L^2} &= \left|\frac{1}{\sqrt{Z}} - \frac{1}{\sqrt{Z_\delta}}\right| \left(\int \mathsf{l}(y|u)\rho(u)\,du\right)^{1/2} \\ &= \frac{|Z - Z_\delta|}{(\sqrt{Z} + \sqrt{Z_\delta})\sqrt{Z_\delta}} \\ &\leq \frac{c_1}{2c_2}\delta,\end{aligned}$$

and

$$\left\|\sqrt{\frac{\mathsf{l}\rho}{Z_\delta}} - \sqrt{\frac{\mathsf{l}_\delta\rho}{Z_\delta}}\right\|_{L^2} = \frac{1}{\sqrt{Z_\delta}}\left(\int \left|\sqrt{\mathsf{l}(y|u)} - \sqrt{\mathsf{l}_\delta(y|u)}\right|^2 \rho(u)\,du\right)^{1/2} \leq \sqrt{\frac{K_1^2}{c_2}}\delta.$$

Therefore

$$\mathsf{D}_{\mathrm{H}}(\pi^y, \pi^y_\delta) \leq \frac{1}{\sqrt{2}}\frac{c_1}{2c_2}\delta + \frac{1}{\sqrt{2}}\sqrt{\frac{K_1^2}{c_2}}\delta = c\delta,$$

with $c = \frac{1}{\sqrt{2}}\frac{c_1}{2c_2} + \frac{K_1}{\sqrt{2c_2}}$, which is independent of $\delta$. □

## 1.4 Model Error

The underlying concept behind this entire section is that the data $y$, which we use to determine unknown parameter $u$, is not actually generated by the identity (1.1). In short the computer code defining forward model $G(\cdot)$ does not exactly represent the physical reality that gave rise to the data, a situation referred to as model error or model misspecification. We describe three approaches to addressing this issue. The starting point for all of them is that the data $y$ arises from noisy observation of a physical process $G_{\mathrm{p}} : \mathbb{R}^d \to \mathbb{R}^k$ so that

$$y = G_{\mathrm{p}}(u) + \eta. \tag{1.11}$$

The assumption is that $G_{\mathrm{p}}$ is not available to us, but that we have a (family of) computational model(s) $G_{\mathrm{c}}$ which we can use in place of $G_{\mathrm{p}}$ to determine $u$ from $y$.

The three approaches we describe, all used in the literature, address the issue of *model error*: namely addressing the fact that $G_{\mathrm{c}}$ does not exactly represent the true physical process $G_{\mathrm{p}}$ generating the data. The reader will appreciate that it is also possible to combine each of the three approaches to devise more general approaches to model error.

### 1.4.1 Representing Error in Data Space

This first approach makes the assumption that $G_{\mathrm{p}} : \mathbb{R}^d \to \mathbb{R}^k$ is related to computer code $G_{\mathrm{c}} : \mathbb{R}^d \to \mathbb{R}^k$ in the sense that

$$G_{\mathrm{p}}(u) = G_{\mathrm{c}}(u) + b \tag{1.12}$$

for some unknown vector $b \in \mathbb{R}^k$. Furthermore, random variable $\eta$ is assumed to be drawn from a centered distribution $\nu_\sigma(\cdot)$, known up to a parameter $\sigma \in \mathbb{R}^p$. Thus model error is present both because of the unknown shift between $G_{\mathrm{p}}$ and $G_{\mathrm{c}}$ and because of the unknown parameter in the distribution of the additive noise.

Rather than just putting a prior $\rho$ on $u$, as in Section 1.1, we put a prior $\rho$ on $(u, b, \sigma)$. The inverse problem is now reformulated as determining the distribution $\nu$ of $(u, b, \sigma)|y$. Combining (1.11) and (1.12) we obtain

$$y = G_{\mathrm{c}}(u) + b + \eta, \tag{1.13}$$

where $\eta \sim \nu_\sigma$. By applying Bayes Theorem 1.2 we obtain

$$\pi^y(u, b, \sigma) = \frac{1}{Z}\nu_\sigma\big(y - G_{\mathrm{c}}(u) - b\big)\rho(u, b, \sigma), \tag{1.14}$$

where

$$Z = Z(y) := \int_{\mathbb{R}^d \times \mathbb{R}^k \times \mathbb{R}^p} \nu_\sigma\big(y - G_{\mathrm{c}}(u) - b\big)\rho(u, b, \sigma)\, du db d\sigma > 0.$$

Remark 1.13. In this setting the prior $\rho(u, b, \sigma)$ is typically factored as an independent product of priors on $u$ and on each of the hyper-parameters $b$ and $\sigma$. ♢

### 1.4.2 Representing Error in Parameter Space

A different approach is to account for model error in parameter space. To achieve this, parameter $u$ is assumed to be perturbed by a draw $\vartheta$ from a random variable with distribution $r_\beta$ known, up to parameter $\beta \in \mathbb{R}^p$. Thus we write

$$y = G(u + \vartheta) + \eta, \tag{1.15}$$

assuming in this setting that $\eta \sim \nu$ and that $\nu$ is completely known.

In principle the parameters $(u, \beta)$ can be found by putting a prior $\rho$ on $(u, \beta)$ and determining the likelihood $\mathsf{l}(y|u, \beta)$ from (1.15). Applying Bayes Theorem 1.2 we obtain

$$\pi^y(u, \beta) = \frac{1}{Z}\mathsf{l}(y|u, \beta)\rho(u, \beta), \tag{1.16}$$

where

$$Z = Z(y) := \int_{\mathbb{R}^d \times \mathbb{R}^p} \mathsf{l}(y|u, \beta)\rho(u, \beta)\, du d\beta > 0.$$

As in Remark 1.13 the prior is usually factored as an independent product, here of $u$ and of $\beta$.

A substantial challenge facing this approach is to define a tractable likelihood $\mathsf{l}(y|u, \beta)$. Rather than restricting to settings where the likelihood is tractable, an alternative approach is to use likelihood-free methods such as approximate Bayesian computation. We will discuss likelihood-free inference in Section 6.2. Here, we illustrate the representation of error in parameter space in the setting of a linear forward model.

**Example 1.14.** For a linear forward model, $G(u) = Au$, the data model (1.15) becomes

$$y = Au + (A\vartheta + \eta).$$

This can be interpreted as a linear forward model with the additive error $A\vartheta + \eta$. If $\vartheta$ and $\eta$ are independent zero-mean Gaussian random variables, the embedded model error corresponds to a likelihood model with inflated variance where the additional variance is related to the forward model. Then $\mathsf{l}(y|u, \beta)$ is determined by

$$\mathbb{P}(y|u, \beta) = \mathcal{N}(Au, AC(\beta)A^\top + \Sigma),$$

where

$$C(\beta) = \mathrm{Cov}(\vartheta), \quad \Sigma = \mathrm{Cov}(\eta).$$

♢

### 1.4.3 Parameterizing the Forward Model

A third approach to model error postulates that the physically realizable forward map $G_\mathrm{p}$ is related to a class of computational models $G_\mathrm{c} : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^k$ in the sense that, for all $u \in \mathbb{R}^d$, and for some $\alpha \in \mathbb{R}^p$,

$$G_\mathrm{p}(u) = G_\mathrm{c}(u, \alpha). \tag{1.17}$$

Remark 1.15. A parameterized forward model may be a coarse-scale model where unresolved small-scale physical processes are described by parameters $\alpha$. For example, $\alpha$ may describe cloud cover in a climate model $G_\mathrm{c}$. More examples will be provided in Chapter 8. ♢

It is assumed that $\alpha$ is not known to us. Thus the problem becomes one of determining the pair $(u, \alpha)$ from the observation $y$. Combining (1.11) and (1.17) suggests consideration of the Bayesian inverse problem of determining the distribution of $(u, \alpha)|y$ given that

$$y = G_\mathrm{c}(u, \alpha) + \eta. \tag{1.18}$$

Here $\eta$ again describes additive noise with known distribution $\nu$. Applying Bayes Theorem 1.2 we obtain

$$\pi^y(u, \alpha) = \frac{1}{Z}\nu(y - G_\mathrm{c}(u, \alpha))\rho(u, \alpha), \tag{1.19}$$

where

$$Z = Z(y) := \int_{\mathbb{R}^d \times \mathbb{R}^p} \nu(y - G_\mathrm{c}(u, \alpha))\rho(u, \alpha)\, dud\alpha > 0.$$

Similarly to Remark 1.13 the prior is usually factored as an independent product of priors on $u$ and on $\alpha$.

## 1.5 Bibliography

For Bayesian approach to inverse problems see [166, 309, 301, 71, 183, 184, 234]. For classical optimization-based approaches to inverse problems, we refer to the books and lecture notes [311, 89, 323, 20, 220]. The concept of MAP estimators, which links probability to optimization, is discussed in the books [166, 309, 72]. For generalizations see the papers [137, 1, 180].

The stability and well-posedness of the Bayesian inverse problem was first studied in [211], using the KL divergence (see Definition 12.27), with respect to perturbations in the data. The articles [301, 71] study similar stability and well-posedness results in the Hellinger metric, as we do here. Related results on stability and well-posedness, but using other distances and divergences, may be found in [185]. The papers [150, 149] discuss generalizations of the well-posedness theory to various classes of specific non-Gaussian priors.

The approach to model error outlined in Subsection 1.4.1 was introduced in [171]. The embedded model error framework of Subsection 1.4.2 was introduced in [279]; for an introduction to approximate Bayesian computation see [289]. An example of the approach to model error described in Subsection 1.4.3 may be found in [64]. Methodologies to identify parameters of structural error models (e.g., within computational models for dynamical processes), often using indirect data, are described in [336, 194].

# Chapter 2

## Variational Inference

The variational formulation of any problem in mathematics is useful because it opens the door to computational methods. Such computational methods seek to minimize the objective function over a strict subset of the whole space over which the original variational problem is posed. If the strict subset if chosen judiciously (this is a problem-dependent choice) then the resulting computational methodology is both tractable and yields a useful, and interpretable, approximation of the solution of the original variational problem posed on the whole space. This chapter is devoted to formulating Bayes Theorem 1.2 variationally and using this formulation as the basis for approximate inference. We make the following data assumption:

**Data Assumption 2.1.** *Data $y \in \mathbb{R}^k$ is given and is assumed to have come from identity (1.1), where $G : \mathbb{R}^d \to \mathbb{R}^k$ is known and $\eta$ is an unknown realization of a centered random variable whose distribution is known and satisfies Assumption 1.1.*

Under the above data assumption, and given a prior distribution on $u$, the posterior distribution on $\mathbb{P}(u|y)$ is defined by Bayes Theorem 1.2. In Section 2.1 we show how this theorem can be formulated variationally as a minimization problem over the set of all probability density functions; this leads to the subject of variational inference, the topic of Section 2.2. In that section we exhibit two choices of subsets of all probability density functions over which to approximate the posterior, the mean-field and Gaussian subsets. This leads to two widely-used methods in variational inference. We conclude the chapter, in Section 2.3, with bibliographical remarks.

## 2.1 Variational Formulation of Bayes Theorem

Our starting point in this section is Bayes Theorem 1.2. Because $y$ dependence is not central to our discussions we write the posterior as $\pi := \pi^y$. A useful formulation of Bayes Theorem arises from seeking the posterior as the solution of an optimization problem over the space of probability density functions. The objective of the optimization problem measures the discrepancy between the posterior density $\pi$ and a candidate density $q$; the objective is constructed to be minimized at $q = \pi$. By using Bayes Theorem 1.2 the objective function can be rewritten in terms of the prior, the likelihood and the normalization constant. In what follows we write $\mathcal{P} = \mathcal{P}(\mathbb{R}^d)$ for the space of all probability density functions over $\mathbb{R}^d$.

One natural class of objective functions $\mathsf{F} : \mathcal{P} \to \mathbb{R}$ is defined by the class of $\mathsf{f}$-divergences $\mathsf{D}_{\mathsf{f}}(q\|\pi)$ :

$$\mathsf{F}(q) := \mathsf{D}_{\mathsf{f}}(q\|\pi) = \int \mathsf{f}\left(\frac{q(u)}{\pi(u)}\right)\pi(u)\,du.$$

See Chapter 12 for the definition and properties of this family of distance-like objects defined on the space of probability densities. By the properties of all divergences, detailed at the start of Section 12.2, we deduce that $\mathsf{F}(q)$ is minimized at $q = \pi$ and that this is the unique minimizer. A particularly useful choice arises from setting $\mathsf{f}(t) = t\log(t)$, leading to definition of the objective $\mathsf{F} : \mathcal{P} \to \mathbb{R}$ given by

$$\mathsf{F}(q) = \mathsf{D}_{\mathrm{KL}}(q\|\pi) = \int \log\left(\frac{q(u)}{\pi(u)}\right)q(u)\,du,$$

the Kullback–Leibler (KL) divergence. This choice of objective is convenient because it can be minimized without knowledge of the normalization constant of $\pi$; see Remark 12.28. Using this KL divergence to formulate variational inference delivers the following theorem:

**Theorem 2.2.** *Consider the Bayesian inverse problem defined by prior $\rho$ and likelihood $\mathsf{l}(y|u)$ given by (1.2), under Assumptions 1.1. Define $\mathsf{J} : \mathcal{P} \to \mathbb{R}$ by*

$$\mathsf{J}(q) = \mathsf{D}_{\mathrm{KL}}(q\|\rho) - \mathbb{E}^q[\log \mathsf{l}(y|u)], \tag{2.1a}$$

$$q_{\mathrm{OPT}} \in \arg\min_{q\in\mathcal{P}} \mathsf{J}(q). \tag{2.1b}$$

*Then $q_{\mathrm{OPT}} = \pi$, the posterior distribution.*

*Proof.* Note that, by Lemma 12.31, it follows that $\mathsf{F}(q) = \mathsf{D}_{\mathrm{KL}}(q\|\pi)$ is non-negative for all inputs $q$ from the set of probability density functions. In addition, by Definition 12.27, it is clear that the infimum of $\mathsf{F}(q)$, namely 0, is attained at $q = \pi$. Furthermore, from Definition 12.1 and Lemma 12.31, it follows that the minimizer is unique since the Hellinger distance, and hence the KL divergence, between two densities is nonzero unless they are identical (strictly speaking in the Lebesgue a.e. sense). Using the form of the posterior density, we see that

$$\mathsf{F}(q) = \mathbb{E}^q[\log q - \log \rho(u) - \log \mathsf{l}(y|u) + \log Z] \tag{2.2a}$$

$$= \mathsf{D}_{\mathrm{KL}}(q\|\rho) - \mathbb{E}^q[\log \mathsf{l}(y|u)] + \log Z \tag{2.2b}$$

$$= \mathsf{J}(q) + \log Z. \tag{2.2c}$$

Because $Z$ is a constant with respect to $q$, the minimizer of $\mathsf{F}(q)$, which is unique, is equivalent to the minimizer of $\mathsf{J}(\cdot)$ and the proof is complete. □

Remark 2.3. Note that, using the loss function (1.5), we may write

$$\mathsf{J}(q) = \mathsf{D}_{\mathrm{KL}}(q\|\rho) + \mathbb{E}^q[\mathsf{L}(u)]. \tag{2.3}$$

Using this formulation of $\mathsf{J}(\cdot)$ we comment on the structure on the minimization problem (2.1) in relation to the MAP estimation problem presented in Section 1.2. The MAP problem requires minimization of (1.7). Comparison with (2.3) shows that in both cases the objective involves two terms that balance the properties of the prior distribution (the first term) and the fit to the data (the second term); furthermore the second term in (2.3) is the expectation of the second term in (1.7). Note that minimization of (2.3) is more general than minimization of (1.7): the former provides a probability density function rather than the point estimator provided by the latter. $\diamondsuit$

Remark 2.4. Because of the desirable property that the objective function can be minimized without knowledge of the normalization constant, we work exclusively with KL divergence to define variational formulations of Bayes Theorem 1.2. While other objective functions with this property do exist, they cannot be $\mathsf{f}$-divergences, by Remark 12.28. $\diamondsuit$

## 2.2 Variational Inference

In practice characterizing the posterior distribution is difficult, in general. One approach to doing so is to approximate the posterior using the variational formulation. We replace the set $\mathcal{P}$ in (2.1) by a tractable class of probability distributions $\mathcal{Q} \subset \mathcal{P}$ for the purpose of computations and for the purpose of revealing an explicit, interpretable form. The variational inference problem is then given by solving the following optimization problem:

$$q^\star \in \arg\min_{q \in \mathcal{Q}} \mathsf{J}(q). \tag{2.4}$$

In the next two subsections we outline two popular classes of tractable distributions $\mathcal{Q}$: in Subsection 2.2.1 the product of independent components for each marginal; and in Subsection 2.2.2 multivariate Gaussians. The former case is known as *mean-field variational inference* and the latter as *Gaussian variational inference.*

### 2.2.1 Mean-Field Family

Mean-field inference works with a family $\mathcal{Q}$ of distributions with independent marginals; that is, the density $q$ factorizes into a product of independent components. While this choice does not represent dependencies among its variables, it is often chosen because of the efficiency of methods for finding $q^\star$ in this case, and because of the ease of computing marginal properties once $q^\star$ is determined.

**Definition 2.5.** A probability density function $q$ for $z \in \mathbb{R}^d$ is in the *mean-field* family $\mathcal{Q}$ if its coordinates are independent. That is, its probability density function can be written in the form

$$q(z) = \prod_{i=1}^{d} q_i(z_i),$$

where $q_i$ is a one-dimensional probability density function for the $i$th coordinates $z_i$. $\diamondsuit$

With this choice of $\mathcal{Q}$ the solution of the optimization problem in (2.10) may be approached by using the following consistency equations, relating the marginal densities

$\{q_i\}_{i=1}^d$ to one another. All integrals appearing in the statement and proof of the following proposition are over the whole of Euclidean space of the relevant dimension.

**Proposition 2.6.** *Let $\mathcal{Q}$ be defined as in Definition 2.5. Then the optimal $q^\star = \prod_{i=1}^d q_i^\star(u_i)$ solving (2.10) satisfies*

$$q_i^\star(u_i) \propto \exp\left(\int \log \pi(u) \prod_{j\neq i} q_j^\star(u_j)\, du_{-i}\right) \qquad \textit{for } i = 1, \dots, d,$$

*where $u_{-i} \in \mathbb{R}^{d-1}$ denotes all variables except for $u_i$.*

*Proof.* Recall that minimizing $\mathsf{J}(\cdot)$ is equivalent to minimizing $\mathsf{D}_{\mathrm{KL}}(\cdot\|\pi)$. The KL divergence $\mathsf{D}_{\mathrm{KL}}(q\|\pi)$ for a density $q$ in the mean-field family is given by

$$\begin{aligned}\mathsf{D}_{\mathrm{KL}}(q\|\pi) &= \int \log\left(\prod_{i=1}^d q_i(u_i)\right) \prod_{i=1}^d q_i(u_i)\, du - \int \log \pi(u) \prod_{i=1}^d q_i(u_i)\, du \\ &= \sum_{i=1}^d \int \log q_i(u_i) q_i(u_i)\, du_i - \int \log \pi(u) \prod_{i=1}^d q_i(u_i)\, du,\end{aligned}$$

where in the last line we used that each integrand only depends on $u_i$ and each $q_i$ is a probability density function that integrates to 1. Taking the first variation of the KL divergence with respect to each $q_i$, we have

$$\frac{\delta}{\delta q_i}\mathsf{D}_{\mathrm{KL}}(q\|\pi) = 1 + \log q_i(u_i) - \int \log \pi(u) \prod_{j\neq i} q_j(u_j)\, du_{-i}.$$

Setting the first variation equal to zero and re-arranging the terms gives us the unnormalized form for each marginal density $q_i^\star$ at any critical point of $\mathsf{J}(\cdot)$ over $\mathcal{Q}$. □

Remark 2.7. We note that although the preceding proposition is formulated in terms of the posterior density, its statement remains unchanged if $\pi(u)$ is replaced by $\rho(u)\mathsf{l}(y|u)$; this simply changes the constant of proportionality. Thus, to use it, we do not need to know the normalization constant $Z$ in Bayes Theorem 1.2. ♢

The proposition defines a set of coupled equations that must be satisfied by the optimal marginal densities in terms of the other marginal densities. Moreover, it prescribes how to set $q_i$ when keeping all other coordinates fixed. Thus, a natural approach to find the minimizer is to perform coordinate updates on each marginal. This algorithm is known as coordinate-ascent variational inference (CAVI). *Ascent* because traditionally formulated in terms of maximizing the $\mathsf{ELBO}$ function (see Definition 2.13) rather than minimizing the KL divergence; we prefer, for consistency with other optimization problems in the notes, to formulate it in terms of $\mathsf{J}$, and hence *descent.*

The updates are thus, sequentially for iteration index $\ell \in \mathbb{Z}^+$ until convergence, and for $i = 1, \dots, d$,

$$q_i^{\ell+1}(u_i) \propto \exp\left(\int \log \pi(u) \prod_{j<i} q_j^{\ell+1}(u_j) \prod_{j>i} q_j^{\ell}(u_j)\, du_{-i}\right). \tag{2.5}$$

**Algorithm 2.1** Coordinate-Ascent Variational Inference

```
**Input**: Density $\pi$ known up to normalizing constant. Initialization $q^0$. Number $L$ of iterations.
For $\ell = 0, 1, 2, \dots, L-1$, compute $q^{\ell+1}$ from $q^\ell$ :
Update $q_i^{\ell+1}$ for $i = 1, \dots, d$ using (2.5).
**Output**: Approximation $q^L$ of density $\pi$.
```

Remark 2.8. The mean-field methodology has the computational advantage of reducing the inference problem in (potentially high) dimension $d$ to one of $d$ independent one-dimensional inference problems. As such it can potentially accurately learn marginal information on specific coordinates; but it cannot learn correlations. In the next subsection we work in a different subset $\mathcal{Q}$, the set of Gaussian measures, in which it is possible to approximate correlation information. ♢

### 2.2.2 Gaussian Distributions

Another tractable variational approach is to seek an approximate distribution $q^\star$ from within a parametric family $\mathcal{Q}$ of probability density functions. In many applications, interest is focused on ultimately estimating the first two moments of $\pi^y$; in this context a natural family to consider is Gaussian distributions. In this subsection we derive optimality conditions for the solution to the resulting optimization problem. To find a Gaussian approximation, we seek to minimize $\mathsf{D}_{\mathrm{KL}}(\cdot\|\pi)$ over the set of distributions

$$\mathcal{Q} := \{q \in \mathcal{P} : q = \mathcal{N}(m, \Sigma), (m, \Sigma) \in \mathcal{C}\}, \tag{2.6a}$$

$$\mathcal{C} := \{m \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d\times d}_{\mathrm{sym},\geq}\}. \tag{2.6b}$$

**Proposition 2.9.** *Let $\mathcal{Q}$ be defined as in (2.6). Then the optimal $q^\star = \mathcal{N}(m^\star, \Sigma^\star)$ solving (2.10) satisfies*

$$(m^\star, \Sigma^\star) \in \arg\min_{(m,\Sigma)\in\mathcal{C}} \mathbb{E}^q[-\log \pi(u)] - \frac{1}{2}\log\det(\Sigma), \tag{2.7}$$

*where the expectation is with respect to $q = \mathcal{N}(m, \Sigma)$.*

*Proof.* Recall that $q^\star$ minimizes $\mathsf{D}_{\mathrm{KL}}(\cdot\|\pi)$ over $\mathcal{Q}$. The result follows from using the fact that, for $q(u)$ being the density of $\mathcal{N}(m, \Sigma)$, we have

$$q(u) = \frac{1}{(2\pi)^{d/2}\det(\Sigma)^{1/2}}\exp\Bigl(-\frac{1}{2}|u-m|_\Sigma^2\Bigr). \tag{2.8}$$

From this it follows that

$$\mathsf{D}_{\mathrm{KL}}(q\|\pi) = \int \log q(u)\, q(u)\, du - \int \log \pi(u)\, q(u)\, du,$$

$$\int \log q(u)\, q(u)\, du = -\frac{d}{2} - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log\det(\Sigma).$$

□

Remark 2.10. The two terms in the objective function can be interpreted as having a competitive behavior. While the second term can be maximized by a Gaussian distribution of increasing smaller variance centered at the mode of $\pi$, the point of highest posterior density, the first term approaches negative infinity if the variance of any marginal approaches zero. Hence, the first term in the objective regularizes the optimization problem to ensure the approximating Gaussian is not degenerate in any direction. $\diamondsuit$

**Proposition 2.11.** *Let $(m^\star, \Sigma^\star) \in \mathcal{C}$ solve the Gaussian variational inference in Proposition 2.9. The solution satisfies the first-order optimality conditions*[1]

$$\mathbb{E}^\xi[D \log \pi(m^\star + (\Sigma^\star)^{1/2}\xi)] = 0, \quad \mathbb{E}^\xi[D^2 \log \pi(m^\star + (\Sigma^\star)^{1/2}\xi)] = -(\Sigma^\star)^{-1},$$

*where $\xi \sim \mathcal{N}(0, I_d)$.*

*Proof.* Throughout the proof, use of $D$ or $D^2$ without a subscript is to be taken as with respect to variable $u$; in addition, $q$ the Gaussian density $\mathcal{N}(m, \Sigma)$ given in (2.8). The first-order optimality condition for the optimization problem in (2.7) is given by

$$D_{(m,\Sigma)} \left[ \mathbb{E}^q[-\log \pi(u)] - \frac{1}{2} \log \det(\Sigma) \right] \bigg|_{(m^\star, \Sigma^\star)} = 0.$$

The gradients of the objective with respect to $m$ and $\Sigma$ are given by

$$\begin{aligned} D_m \left[ \int -\log \pi(u) q(u)\, du - \frac{1}{2} \log \det(\Sigma) \right] &= \int -\log \pi(u) D_m q(u)\, du, \\ D_\Sigma \left[ \int -\log \pi(u) q(u)\, du - \frac{1}{2} \log \det(\Sigma) \right] &= \int -\log \pi(u) D_\Sigma q(u)\, du - \frac{1}{2}\Sigma^{-1}. \end{aligned}$$

From the symmetry of the Gaussian density with respect to $u$ and $m$, we have

$$\begin{aligned} D_m q(u) &= D q(u), \\ D_\Sigma q(u) &= q(u) D_\Sigma \log q(u) \\ &= -q(u) \frac{1}{2} [\Sigma^{-1} - \Sigma^{-1}(u-m)(u-m)^\top \Sigma^{-1}] \\ &= \frac{1}{2} D^2 q(u). \end{aligned}$$

Substituting these expressions above and applying integration by parts gives us

$$\begin{aligned} \int -\log \pi(u) D_m q(u)\, du &= \int -\log \pi(u) D q(u)\, du = \int D \log \pi(u) q(u)\, du, \\ \int -\log \pi(u) D_\Sigma q(u)\, du &= -\frac{1}{2} \int \log \pi(u) D^2 q(u)\, du = -\frac{1}{2} \int D^2 \log \pi(u) q(u)\, du. \end{aligned}$$

Setting the two gradients equal to zero and re-writing the expectations in terms of a standard Gaussian random variable $\xi \sim \mathcal{N}(0, I_d)$ using the relation $u = m + \Sigma^{1/2}\xi$ for a positive definite covariance $\Sigma$ gives us the result. $\square$

Remark 2.12. If $c^- I_d \preceq D^2 \log \pi(u) \preceq c^+ I_d$ for all $u \in \mathbb{R}^d$, then the optimal solution satisfies $\frac{1}{c^+} I_d \preceq \Sigma^\star \preceq \frac{1}{c^-} I_d$. $\diamondsuit$

[1] Use of $D$ and $D^2$ here denote derivatives with respect to $u$, then evaluated at $u = m^\star + (\Sigma^\star)^{1/2}\xi$.

### 2.2.3 Mode-Seeking Versus Mean-Seeking Variational Inference

In the previous subsections, we seek an approximation to the posterior distribution $\pi$ by minimizing, over $\mathcal{Q} \subset \mathcal{P}$, the function

$$q \mapsto \mathsf{D}_{\mathrm{KL}}(q\|\pi) = \mathbb{E}_q[\log(q/\pi)];$$

this is sometimes referred to as the reverse KL divergence. Using the reverse KL divergence as an objective is computationally convenient in the setting where we can evaluate the likelihood and the prior, but we do not necessarily know the normalization constant, nor do we necessarily have the ability to easily sample from the posterior. In comparison, the forward KL divergence

$$q \mapsto \mathsf{D}_{\mathrm{KL}}(\pi\|q) = \mathbb{E}_\pi[\log(\pi/q)]$$

is more convenient for optimization over $\mathcal{Q}$ when we can sample from $\pi$, but do not necessarily have access to an analytical expression for the target density.

In addition to computational considerations, the two choices of KL divergence lead to different behavior when minimizing over a subset of probability measures. The reverse KL favors approximate distributions where $\log(q/\pi)$ is small in regions of high probability of $q$. This occurs when $q \approx \pi$ for large $q$ or when $q \ll \pi$. As a result, the minimizers for $q$ tend to fit one mode of multi-modal distributions $\pi$, but miss the other modes. Hence, minimizing the reverse KL is known is also known as *mode-seeking*. In contrast, the minimizers of the forward KL favor approximate distributions $q$ where $\log(\pi/q)$ is small in regions of high probability of $\pi$. This occurs by having $q$ be non-zero everywhere in the support of $\pi$ so the denominator in the log does not approach zero. Placing mass everywhere in the support of $\pi$ leads to a *mean-seeking* behavior, where the minimizer for $q$ corresponds to matching the moments of the target density $\pi$. In particular, when $\mathcal{Q}$ comprises Gaussians the minimizer of the forward KL divergence is given by Gaussian $q$ with the same mean and covariance as those moments under $\pi$. Further discussion on these topics and references to the literature may be found in the bibliography in Section 2.3.

### 2.2.4 Evidence Lower Bound

We now relate $\mathsf{J}(q)$, defined by (2.1a), to an important concept in variational inference:

**Definition 2.13.** The *evidence lower bound* (ELBO) of a probability density $q$ with respect to an unnormalized density $\widetilde{\pi}$ is

$$\mathsf{ELBO}(\widetilde{\pi}, q) = \mathbb{E}^q[\log \widetilde{\pi}(u)] - \mathbb{E}^q[\log q(u)].$$

$\diamondsuit$

The reason for the terminology evidence lower bound will be made clear through Remarks 2.15, 2.16. The evidence lower bound is sometimes also known as the negative variational free energy. The definition is often used in the situation where $\widetilde{\pi}(u)$ is found from the joint distribution of random variable $(u, y)$, with $y$ frozen. This is the setting of the following proposition, which sheds first light on the significance of the ELBO functional.

**Proposition 2.14.** *Define the unnormalized density $\widetilde{\pi}(u) = \rho(u)\mathsf{l}(y|u)$ and define the normalization constant $Z$ by $\pi = Z^{-1}\widetilde{\pi}$, assuming that $Z > 0$. Then, the maximizer*

$$q_{\text{OPT}} \in \arg\max_{q\in\mathcal{P}} \mathsf{ELBO}(\widetilde{\pi}, \cdot)$$

*is attained at the posterior distribution $q_{\text{OPT}} = \pi$. Furthermore, the maximum value is* $\mathsf{ELBO}(\widetilde{\pi}, q_{\text{OPT}}) = \log(Z)$.

*Proof.* This follows from recognizing from (2.2a) that, with the stated form of $\widetilde{\pi}$,

$$\mathsf{ELBO}(\widetilde{\pi}, q) = -\mathsf{D}_{\text{KL}}(q\|\pi) + \log(Z) = -\mathsf{J}(q). \tag{2.9}$$

At optimality we have $\mathsf{D}_{\text{KL}}(q_{\text{OPT}}\|\pi) = 0$, and so $\mathsf{ELBO}(\pi, q_{\text{OPT}}) = \log(Z)$. □

Remark 2.15. Notice that $\widetilde{\pi}(u) = \mathbb{P}(u, y)$ and that the normalizing constant $Z = \mathbb{P}(y)$ is called the *evidence*. Thus the preceding proposition demonstrates the significance of the quantity $\mathsf{ELBO}(\pi, q_{\text{OPT}})$: it computes the logarithm of the evidence, namely the log-probability that the observed data $y$ was produced by the proposed statistical model for the joint random variable $(u, y)$. ♢

Recall that variational inference is performed by solving the following optimization problem over $\mathcal{Q} \subset \mathcal{P}$ :

$$q^\star \in \arg\min_{q\in\mathcal{Q}} \mathsf{J}(q). \tag{2.10}$$

Remark 2.16. The calculations in the proof of Proposition 2.14 show that, using (2.9),

$$\log(Z) = \mathsf{ELBO}(\widetilde{\pi}, q_{\text{OPT}}) \geq \mathsf{ELBO}(\widetilde{\pi}, q^\star) = -\mathsf{J}(q^\star).$$

Hence, after variational inference over $\mathcal{Q}$ is performed, we can estimate the normalizing constant, or the evidence, by $Z \geq \exp\bigl(-\mathsf{J}(q^\star)\bigr)$. The terminology *evidence lower bound* is now made clear. ♢

## 2.3 Bibliography

The paper [338] provided impetus for variational Bayes, highlighting the fact that the posterior is found by minimizing $\mathsf{J}(\cdot)$ given in (2.10). Our presentation has focused on finding the posterior as the minimizer of the KL divergence, but there are other functionals that may be considered. For instance, [106] showed that the posterior is found by minimizing a functional of the same structural form as (2.2) based on the $\chi^2$ divergence (see Definition 12.27), rather than KL divergence. [197, 139] showed how to perform variational inference using the family of Renyi-alpha divergences, which include the KL and $\chi^2$ divergences as well as the Hellinger metric. Whilst there are many divergences that could be used to define Bayes Theorem via optimization over the space of measures, the specific choice of forward KL divergence has a special place: amongst a wide class of divergences it is the unique choice for which the objective function does not require knowledge of the normalization constant $Z$ [58].

Refer to [146] for a modern approach to variational inference, with applications to large collections of documents and topic models. For applications in graphical models see [329]. An extension to the mean-field model is to include interactions [165]. Natural extensions of Gaussians include Gaussian mixtures in [178]. The derivation of the mean-field equations for Gaussians was first shown in [240]. Discussion of mean-seeking versus mode-seeking may be found in [278]; see Sections 4.2 and 4.3 in particular.

The ELBO objective is commonly used to learn variational auto-encoders [174]. In this context, the goal is to find an approximate distribution $q(u) = \int p_\theta(u|z)p(z)\,dz$, for which it is intractable to evaluate the KL divergence. By working with the ELBO, we can maximize a lower bound for the negative KL divergence. The maximum value corresponds to the marginal likelihood for the data under $q$.

The ELBO is a non-convex functional even for simple potentials; see the example in Appendix G of [179]. Due to this non-convexity, it is often challenging to develop guarantees for VI outside of restricted settings on the distribution. Recently, there have been many statistical and algorithmic guarantees for variational inference. We refer to [178] for analysis using gradient flows.

# Chapter 3

## Forward Surrogate Modelling

In this chapter our focus is on solving the inverse problem for $u$ given $y$, defined by (1.1). The key idea is to approximate $G : \mathbb{R}^d \to \mathbb{R}^k$ with a cheap-to-evaluate surrogate, $G_\delta$, using ideas of supervised learning from Chapter 14. Thus, to learn this approximation, we make the following assumption.

**Data Assumption 3.1.** *Data is available in the form*

$$\left\{u^{(n)}, y^{(n)}\right\}_{n=1}^{N}, \tag{3.1}$$

*where the* $\{u^{(n)}\}_{n=1}^{N}$ *are generated i.i.d. from probability density function* $\Upsilon \in \mathcal{P}(D)$*, for some* $D \subseteq \mathbb{R}^d$*, and where* $y^{(n)} = G(u^{(n)})$*.*

Remark 3.2. A key point to appreciate is that, since we will use approximation of $G$ to solve the Bayesian inverse problem for $\pi^y$ given by Bayes Theorem 1.2, an ideal choice for $\Upsilon$ is that it is close to $\pi^y$. This, of course, leads to a chicken-egg issue. In practice this can be addressed by choosing $\Upsilon$ to have generous support, aiming to subsume that of $\pi^y$; or by generating data in tandem with solving the inverse problem. ♢

In Section 3.1 we discuss the use of surrogate forward models to speed-up Bayesian inversion. Section 3.2 provides analysis of the effect of approximating the forward model on the posterior. We briefly comment on surrogate modelling in the context of MAP estimation when model error is also being inferred in Section 3.3, slightly generalizing Data Assimilation 3.1. We conclude in Section 3.4 with bibliographic remarks.

## 3.1 Accelerating Bayesian Inversion

Recall the inverse problem (1.1) of finding $u$ from $y$ where

$$y = G(u) + \eta,$$

under the setting summarized in Assumption 1.10. If the Bayesian approach is adopted and Markov chain Monte Carlo (MCMC) is used to sample the posterior, generating each new sample typically requires evaluating the likelihood, and hence the forward model $G$. Even when MAP estimation is used, multiple evaluations of $G$ may be needed to optimize the objective function; if model error is being jointly learned, in the setting of Subsection 1.4.3, then multiple MAP estimators may be required leading to many more evaluations. Thus, when $G$ is computationally expensive to evaluate, multiple evaluation may be prohibitive for both MCMC sampling and MAP estimation. We address this issue by using cheap computational surrogates $G_\delta$.

The methods described in Chapter 14 can be used to approximate the (scalar-valued) likelihood $\mathsf{l}$ resulting from $G$, by an approximation $\mathsf{l}_\delta$ with small error $\delta$, uniformly over bounded open $D \subset \mathbb{R}^d$. In Chapter 14 we focus on learning functions taking values in $\mathbb{R}$. However all such methods can be generalized to approximate (vector-valued) $G$ by a uniform approximation over $D \subset \mathbb{R}^d$, $G_\delta$, which in turn results in a uniform approximation $\mathsf{l}_\delta$ of the likelihood. Such an approximation $G_\delta$ can be learned from Data Assumption 3.1. If carefully designed, then the machine learning approximation of the likelihood will be much faster to evaluate than the true likelihood and, because of the approximation properties, will result in accurate posterior inference, with errors of size $\delta$. Such results rely on error bounds of the form given in Section 14.4 which (with high probability) can be obtained with a number $N$ of evaluations of the forward model which is often orders of magnitude smaller than the number of evaluations required within MCMC, or even MAP estimation in the context of learning model error. The resulting method can thus be very efficient.

## 3.2 Posterior Approximation Theorem

We now establish that the approximate posterior resulting from approximating the likelihood is indeed close to the true posterior. We use error bounds of the form (14.23) within a modification of the well-posedness theory from Chapter 1.

For notational convenience we drop the dependence of the likelihood on $y$ in the remainder of this chapter. Specifically we let

$$\mathsf{l}(u) = \nu(y - G(u)) \quad \text{and} \quad \mathsf{l}_\delta(u) = \nu(y - G_\delta(u))$$

denote the true and approximate likelihoods. Thus we may define the true and approximate posterior distributions by

$$\pi^y(u) = \frac{1}{Z}\mathsf{l}(u)\rho(u) \quad \text{and} \quad \pi^y_\delta(u) = \frac{1}{Z_\delta}\mathsf{l}_\delta(u)\rho(u);$$

here $Z, Z_\delta$ are the corresponding normalizing constants. To prove our result about the closeness, between the true posterior and the posterior with machine-learned likelihood, we specify our assumptions.

**Assumption 3.3.** *The prior distribution on $u$ with density $\rho$ is supported on bounded open set $D \subset \mathbb{R}^d$ and the data defined by (1.1) and Assumption 1.10 has positive probability under the resulting joint distribution on $(u, y)$, so that $Z > 0$. Furthermore, there exist $\delta^+ > 0$ and $K_1, K_2 \in (0, \infty)$ such that, for all $\delta \in (0, \delta^+)$,*

*(i)* $\sup_{u \in D} |\sqrt{\mathsf{l}(u)} - \sqrt{\mathsf{l}_\delta(u)}| \le K_1\delta;$

*(ii)* $\sup_{u \in D}(|\sqrt{\mathsf{l}(u)}| + |\sqrt{\mathsf{l}_\delta(u)}|) \le K_2.$

Parts (i) and (ii) of the assumptions follow from approximation theoretic results of the form given in Chapter 14 on supervised learning, and (14.23) in particular. Note also that combining (i) and (ii) gives

$$\sup_{u\in D} |\mathsf{l}(u) - \mathsf{l}_\delta(u)| \le K_1 K_2 \delta. \tag{3.2}$$

Using the assumption we may state the following approximation result for the machine-learned posterior. When combined with Lemma 12.4, the following theorem guarantees that expectations computed with respect to the machine-learned posterior commit an error of order $\delta$, the magnitude of the error in the approximate likelihood.

**Theorem 3.4.** *Under Assumptions 1.1 and 3.3 there exists $c \in (0, +\infty)$ and $\Delta > 0$ such that, for all $\delta \in (0, \Delta)$,*

$$\mathsf{D}_{\mathrm{H}}(\pi^y, \pi^y_\delta) \le c\delta.$$

As in the proof of the well-posedness Theorem 1.11, we first show a lemma which characterizes the normalization constants and their approximation in the small $\delta$ limit. In the lemma, and the proof of theorem that follows it, all integrals are over the set $D$ defined in Assumption 3.3.

**Lemma 3.5.** *Under Assumptions 1.1 and 3.3 there exist $\Delta > 0$, $c_1, c_2 \in (0, +\infty)$ such that, for all $\delta \in (0, \Delta)$,*

$$|Z - Z_\delta| \le c_1\delta \quad \textit{and} \quad Z, Z_\delta \ge c_2.$$

*Proof.* Since $Z = \int \mathsf{l}(u)\rho(u)\,du$ and $Z_\delta = \int \mathsf{l}_\delta(u)\rho(u)\,du$, we have, using (3.2),

$$\begin{aligned} |Z - Z_\delta| &= \left| \int \big(\mathsf{l}(u) - \mathsf{l}_\delta(u)\big)\rho(u)\,du \right| \\ &\le \int |\mathsf{l}(u) - \mathsf{l}_\delta(u)|\rho(u)\,du \\ &\le K_1K_2\delta, \quad \delta \in (0, \delta^+). \end{aligned}$$

Therefore, for $\delta \le \Delta := \min\{\frac{Z}{2K_1K_2}, \delta^+\}$, we have

$$Z_\delta \ge Z - |Z - Z_\delta| \ge \frac{1}{2}Z.$$

The lemma follows by taking $c_1 = K_1K_2$ and $c_2 = \frac{1}{2}Z$, and noting that $K_1, K_2, Z > 0$ by assumption.

□

*Proof of Theorem 3.4.* Following proof of the well-posedness Theorem 1.11, we decompose the total error into two contributions, reflecting respectively the difference between $Z$ and $Z_\delta$, and the difference between the likelihoods $\mathsf{l}$ and $\mathsf{l}_\delta$:

$$\mathsf{D}_{\mathrm{H}}(\pi^y, \pi^y_\delta) \le \frac{1}{\sqrt{2}} \left\| \sqrt{\frac{\mathsf{l}\rho}{Z}} - \sqrt{\frac{\mathsf{l}\rho}{Z_\delta}} \right\|_{L^2} + \frac{1}{\sqrt{2}} \left\| \sqrt{\frac{\mathsf{l}\rho}{Z_\delta}} - \sqrt{\frac{\mathsf{l}_\delta\rho}{Z_\delta}} \right\|_{L^2}.$$

Using Lemma 3.5 we have, for $\delta \in (0, \Delta)$,

$$\begin{aligned}\left\|\sqrt{\frac{\mathsf{l}\rho}{Z}} - \sqrt{\frac{\mathsf{l}\rho}{Z_\delta}}\right\|_{L^2} &= \left|\frac{1}{\sqrt{Z}} - \frac{1}{\sqrt{Z_\delta}}\right| \left(\int \mathsf{l}(u)\rho(u)\, du\right)^{1/2}\\ &= \frac{|Z - Z_\delta|}{(\sqrt{Z} + \sqrt{Z_\delta})\sqrt{Z_\delta}}\\ &\le \frac{c_1}{2c_2}\delta,\end{aligned}$$

and

$$\left\|\sqrt{\frac{\mathsf{l}\rho}{Z_\delta}} - \sqrt{\frac{\mathsf{l}_\delta\rho}{Z_\delta}}\right\|_{L^2} = \frac{1}{\sqrt{Z_\delta}}\left(\int \left|\sqrt{\mathsf{l}(u)} - \sqrt{\mathsf{l}_\delta(u)}\right|^2 \rho(u)\, du\right)^{1/2} \le \sqrt{\frac{K_1^2}{c_2}}\delta.$$

Therefore

$$\mathsf{D}_{\mathrm{H}}(\pi^y, \pi^y_\delta) \le \frac{1}{\sqrt{2}}\frac{c_1}{2c_2}\delta + \frac{1}{\sqrt{2}}\sqrt{\frac{K_1^2}{c_2}}\delta = c\delta,$$

with $c = \frac{1}{\sqrt{2}}\frac{c_1}{2c_2} + \frac{1}{\sqrt{2}}\sqrt{\frac{K_1^2}{c_2}}$ independent of $\delta$. □

## 3.3 Accelerating MAP Estimation

This section is centred on the approach to model error described in Subsection 1.4.3. For this purpose, recall the Bayesian inverse problem to jointly estimate unknown parameter $u$ and unknown model error parameter $\alpha$ defined in (1.19). Thus we have access to a parameterized family of computational models $G_{\mathrm{c}}(\cdot; \alpha) : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^k$, for $\alpha \in \mathbb{R}^p$. Our data assumption is as follows:

**Data Assumption 3.6.** *Data is available in the form*

$$\left\{u^{(n)}, \alpha^{(n)}, y^{(n)}\right\}_{n=1}^N, \tag{3.3}$$

*where the $\{u^{(n)}, \alpha^{(n)}\}_{n=1}^N$ are generated i.i.d. from probability density function $\Upsilon \otimes q \in \mathcal{P}(D \times \mathbb{R}^p)$, for some $D \subseteq \mathbb{R}^d$, and where $y^{(n)} = G_{\mathrm{c}}(u^{(n)}, \alpha^{(n)})$.*

From this we may learn a cheap surrogate for $G_{\mathrm{c}}$, using the techniques of Chapter 14. This may be used to accelerate Bayesian inference for $\mathbb{P}(u, \alpha|y)$, similarly to what is outlined in the previous section. Here we provide another example of the value of surrogate modeling by focussing, instead, on the MAP estimation problem.

The MAP estimation problem requires minimization, over the pair $(u, \alpha)$, of

$$\mathsf{J}(u, \alpha) = -\log \nu(y - G_{\mathrm{c}}(u, \alpha)) - \log \rho(u, \alpha).$$

Some methods for this problem work by alternating minimization over $u$ and over $\alpha$; a prototypical such method is to iterate, for $\ell$ until convergence, starting from initial

$\alpha^0 \in \mathbb{R}^p$ :

$$u^{\ell+1} \in \arg\min_{u \in \mathbb{R}^d} \mathsf{J}(u, \alpha^\ell),$$
$$\alpha^{\ell+1} \in \arg\min_{\alpha \in \mathbb{R}^p} \mathsf{J}(u^{\ell+1}, \alpha).$$

For each $\ell$, multiple evaluations of $G_{\mathrm{c}}(\cdot, \cdot)$ may be required and use of a surrogate may make the method more efficient.

## 3.4 Bibliography

Using emulators to speed up forward model evaluations, for example in the context of likelihood evaluation in Bayesian inversion, was first introduced as a systematic methodology in [273], and taken further in the realm of Bayesian model error estimation in [171]. The paper [300] studies the use of Gaussian processes for emulation, and derives errors bounds quantifying the effect of emulation error on the posterior. The methodology is developed for a range of applications in the geosciences in [62]; in particular that paper addresses the issue of how to determine $\Upsilon$ in tandem with solution of the inverse problem as discussed in Remark 3.2. A specific application of the idea in climate science may be found in [85]. Data-driven discretizations of forward models for Bayesian inversion are studied in [29]. A recent approach to directly learning parameter to solution (forward) and solution to parameter (inverse) surrogate maps may be found in [318].

An important aspect of learning forward models is to choose the pair of supervised training data over which we would like to find an accurate surrogate model. While we would ideally like to be accurate over the support of the posterior, finding such a surrogate model requires being able to sample the posterior. Instead, it is common to seek the approximate surrogate model to be accurate over the support of the prior. To address this, several recent methods use approximate posteriors to iteratively refine the approximation of the surrogate model [62, 138].

# Chapter 4

## Learning Prior and Regularizers

In Bayesian inference the prior acts as a form of regularization. This can be seen explicitly in two ways. First, by considering MAP estimation as described in Section 1.2, where the objective function to be minimized is the sum of two terms, one of which, the regularizer, is the negative logarithm of the prior $\rho$. Second, by considering the variational form of Bayes theorem as described in Remark 2.3, where the posterior minimizes an objective over probability densities comprising two terms, one of which, the regularizer, is the KL divergence between the putative minimizer $q$ and the prior $\rho$. However, defining a prior from data can be challenging. Often modellers make simple choices, such as Gaussians, because this limits the space of potential regularizers to consider. Here we consider data-driven approaches to determining the prior.

In Section 4.1 we discuss learning the prior measure from data, relating this problem to the unsupervised learning task studied in Chapter 13. Then, in Section 4.2, we discuss representing the prior via a pushforward, a setting that often arises when the prior is learned from data as a transport. Section 4.3 contains a theoretical analysis of the effect of errors in the prior on errors in the posterior. Section 4.4 is devoted to finding regularizers for MAP estimation. This topic is generalized in Section 4.5 to a probabilistic setting. Section 4.6 contains concluding bibliographic remarks. We make three separate data assumptions: Data Assumption 4.1 in Sections 4.1 and 4.3; Data Assumption 4.8 in Section 4.4 and Data Assumption 4.9 in Section 4.5.

## 4.1 Learning the Prior

Recall the Bayesian inverse problem of finding $u \in \mathbb{R}^d$ from $y \in \mathbb{R}^k$ when related by (1.1), so that

$$y = G(u) + \eta.$$

Under the assumptions of Theorem 1.2, the posterior distribution is given by (1.3):

$$\pi(u) = \frac{1}{Z}\nu(y - G(u))\rho(u); \tag{4.1}$$

we have dropped the explicit dependence of the posterior on $y$ for notational convenience. In this section we describe the idea of learning the prior from data. We make the following data assumption to enable this:

**Data Assumption 4.1.** *We are given samples $\{u^{(n)}\}_{n=1}^N$ assumed to be drawn i.i.d. from prior measure $\rho$ on $u$ which is unknown.*

Compare this to Data Assumption 13.1 in Chapter 13, which arises in unsupervised learning, and notice that both assumptions are identical by taking $\Upsilon := \rho$. Thus in this chapter we assume that the prior is only given to us through data, and we may seek to approximate it using the generative modelling techniques of Chapter 13. To clearly understand the methodology of this section it is important to distinguish between the one piece of data, $y$, for which we wish to solve the inverse problem defined by (1.1), and the *training data* $\{u^{(n)}\}_{n=1}^N$ which we assume are available to us, and which we use to learn about the prior.

## 4.2 Representing the Prior via a Pushforward

Our focus is on the posterior distribution $\pi$ defined from the prior $\rho$ by (4.1):

$$\pi(u) = \frac{1}{Z}\nu(y - G(u))\rho(u).$$

When the prior $\rho$ is only known through an empirical approximation $\rho^N \approx \rho$ it is possible to use ideas from transport, described in Chapter 13, to find map $g$ that (approximately) pushes forward a given density $\zeta$ on latent space $\mathbb{R}^{d_z}$ into the prior; that is, $\rho \approx g_\sharp \zeta$. In the remainder of this section we assume that $\rho = g_\sharp \zeta$, noting that the effects of approximating this identity will be discussed in the next section. Regarding the pushforward representation of the prior we have the following useful result:

**Proposition 4.2.** *Let $d_z = d$ and assume that $\rho = g_\sharp \zeta$ and that $g$ is invertible. Define*

$$\pi_g(z) = \frac{1}{Z}\nu\Big(y - G\big(g(z)\big)\Big)\zeta(z). \tag{4.2}$$

*It follows that $\pi = g_\sharp \pi_g$.*

Remark 4.3. The significance of Proposition 4.2 is that it implies that we can solve the Bayesian inverse problem (4.2), posed in the latent space, and push forward under $g$ to find solutions of the original Bayesian inverse problem defined by (4.1). For example if we generate samples under $\pi_g$ in the latent space, then application of $g$ to those samples will generate samples under $\pi$ in the original space.

Hence, if pair $(g, \zeta)$ are known, then the original Bayesian inverse problem for $\pi$ on $\mathbb{R}^d$ may be converted to one for $\pi_g$ on $\mathbb{R}^{d_z}$; pushforward of $\pi_g$ under $g$ yields solution to the original problem. A similar expression to (4.2) may be established even if $d_z < d$, but it no longer follows that $\pi = g_\sharp \pi_g$ because $g$ is not invertible. ♢

*Proof of Proposition 4.2.* From Lemma 12.33 with $u = g(z)$, and (12.16) in particular,

$$\begin{aligned} g_\sharp \pi_g(u) &= (\pi_g \circ g^{-1})(u) \mathrm{det} D(g^{-1})(u) \\ &= \frac{1}{Z} \nu(y - G(u)) (\zeta \circ g^{-1})(u) \mathrm{det} D(g^{-1})(u) \\ &= \frac{1}{Z} \nu(y - G(u)) g_\sharp \zeta(u) \\ &= \frac{1}{Z} \nu(y - G(u)) \rho(u), \\ &= \pi(u) \end{aligned}$$

as required. □

## 4.3 Perturbations to the Prior

Remark 4.3 shows that, if we learn an exact transport map $g$ satisfying $\rho = g_\sharp \zeta$, then we have access to the true posterior with prior $\rho$. In practice, however, we learn $g$ from $\rho^N \approx \rho$ and so we cannot hope to recover an exact transport map; the issue of lack of exactness is further compounded by only learning the transport map over a parametric family of densities and by not iterating the optimization solver to convergence. In Subsections 4.3.1 and 4.3.2 we prove two theorems which address this effect. In Theorem 4.5 we assume that the approximation of the exact transport map leads to an approximate smooth prior, in the space of $u$, and look at the effect on the posterior on the distance between this prior and the true prior. In Theorem 4.6 we study a related problem, namely the effect on the posterior of replacing the prior by an empirical approximation.

### 4.3.1 Smooth Approximation of the Prior

Consider the inverse problem arising for use of a smooth approximate prior $\rho' \approx \rho$. This gives rise to an approximate posterior, $\pi'$ :

$$\pi'(u) = \frac{1}{Z'} \nu(y - G(u)) \rho'(u). \tag{4.3}$$

It is thus important to know that a small change in the prior leads to a small change in the posterior. We prove this in the following theorem which exhibits conditions under which the prior to posterior map is Lipschitz in the Hellinger metric. In the proof of the theorem, and conditions that precede it, we let $\mathsf{l}(u) = \nu(y - G(u))$, suppressing dependence on $y$.

**Assumption 4.4.** *The prior distributions $\rho$ and $\rho'$ are both supported on bounded open set $D \subset \mathbb{R}^d$. There exists $K \in (0, \infty)$ such that $\sup_{u \in D} \mathsf{l}(u) = K$.*

**Theorem 4.5.** *Let Assumption 4.4 hold. Consider posteriors $\pi, \pi'$ in (4.1), (4.3) corresponding, respectively, to priors $\rho, \rho'$. Then, we have*

$$\mathsf{D}_{\mathrm{H}}(\pi, \pi') \le \left( \frac{2K}{(\sqrt{Z} + \sqrt{Z'})\sqrt{Z'}} + \frac{\sqrt{K}}{\sqrt{Z'}} \right) \mathsf{D}_{\mathrm{H}}(\rho, \rho'). \tag{4.4}$$

*Proof.* In the proof all $L^2$ norms are over domain $D$ and all integrals are restricted to domain $D$. With this notation we have that

$$\begin{aligned}\mathsf{D}_{\mathrm{H}}(\pi,\pi') &= \frac{1}{\sqrt{2}}\left\|\sqrt{\frac{\mathsf{l}\rho}{Z}}-\sqrt{\frac{\mathsf{l}\rho'}{Z'}}\right\|_{L^2}\\ &\le \frac{1}{\sqrt{2}}\left\|\sqrt{\frac{\mathsf{l}\rho}{Z}}-\sqrt{\frac{\mathsf{l}\rho}{Z'}}\right\|_{L^2}+\frac{1}{\sqrt{2}}\left\|\sqrt{\frac{\mathsf{l}\rho}{Z'}}-\sqrt{\frac{\mathsf{l}\rho'}{Z'}}\right\|_{L^2}\\ &= \frac{1}{\sqrt{2}}\left|\frac{1}{\sqrt{Z}}-\frac{1}{\sqrt{Z'}}\right|\sqrt{Z}+\frac{1}{\sqrt{2}\sqrt{Z'}}\|\sqrt{\mathsf{l}\rho}-\sqrt{\mathsf{l}\rho'}\|_{L^2}.\end{aligned}$$

The first term can be written as

$$\left|\frac{1}{\sqrt{Z}}-\frac{1}{\sqrt{Z'}}\right|\sqrt{Z}=\frac{|Z-Z'|}{(\sqrt{Z}+\sqrt{Z'})\sqrt{Z'}}.$$

The difference between the normalization constants can be bounded, using Lemma 12.2, by

$$\begin{aligned}|Z-Z'| &\le \int |\mathsf{l}(u)\rho(u)-\mathsf{l}(u)\rho'(u)|\,du\\ &\le K\int |\rho(u)-\rho'(u)|\,du\\ &= 2K\mathsf{D}_{\mathrm{TV}}(\rho,\rho')\\ &\le 2K\sqrt{2}\mathsf{D}_{\mathrm{H}}(\rho,\rho').\end{aligned}$$

For the second term, we have

$$\|\sqrt{\mathsf{l}\rho}-\sqrt{\mathsf{l}\rho'}\|_{L^2}\le\sqrt{K}\|\sqrt{\rho}-\sqrt{\rho'}\|_{L^2}=\sqrt{2K}\mathsf{D}_{\mathrm{H}}(\rho,\rho').$$

Collecting the bounds for the two terms, we obtain the result. □

### 4.3.2 Empirical Approximation of the Prior

Theorem 4.5 applies with priors $\rho,\rho'$ that have a probability density function. Here we consider the setting where the approximate prior is specified empirically by a collection random samples $u^{(n)}$ for $n=1,\dots,N$ from the true prior $\rho$. Then we have the *random probability measure*

$$\rho^N(u)=\frac{1}{N}\sum_{n=1}^{N}\delta(u-u^{(n)}).$$

With this prior we obtain posterior

$$\pi^N(u)=\frac{1}{Z}\nu(y-G(u))\rho^N(u). \tag{4.5}$$

We wish to compare $\pi$ with $\pi^N$. In this setting, it is natural to use the following metric from Subsection 12.1.6:

$$d(\pi,\pi^N)=\sup_{|f|_\infty\le 1}\left|\mathbb{E}\Big[\big(\pi(f)-\pi'(f)\big)^2\Big]\right|^{1/2}, \tag{4.6}$$

where $\pi(f) := \mathbb{E}^{u\sim\pi}[f(u)]$ and $\pi^N(f) := \mathbb{E}^{u\sim\pi^N}[f(u)]$. We note, because it will be useful in what follows, that

$$\sup_{|f|_\infty \le F} \left| \mathbb{E}\Big[ \Big( \pi(f) - \pi^N(f) \Big)^2 \Big] \right|^{1/2} \le F\, d(\pi, \pi^N). \tag{4.7}$$

Recall that this reduces to the total variation metric for non-random measures as discussed in Remark 12.23; and that, furthermore, the square root of the total variation metric upper bounds the Hellinger metric, by Lemma 12.2.

The following proposition uses this metric on random probability measures to quantify the distance between the true and approximate posterior when one prior is specified using samples:

**Theorem 4.6.** *Let Assumption 4.4 hold and let $\pi, \pi^N$ be the posteriors given by (4.1) and (4.5) respectively. Then,*

$$d(\pi, \pi^N) \le \frac{2K}{Z} d(\rho, \rho^N). \tag{4.8}$$

*Proof.* For the true and approximate posterior, we can write the integrals required to estimate $d(\pi, \pi^N)$, with distance given in (4.6), by

$$\pi(f) = \frac{\rho(\mathsf{l}f)}{\rho(\mathsf{l})}, \quad \pi^N(f) = \frac{\rho^N(\mathsf{l}f)}{\rho^N(\mathsf{l})}.$$

Then we have

$$\begin{aligned} \pi(f) - \pi^N(f) &= \frac{\rho(\mathsf{l}f)}{\rho(\mathsf{l})} - \frac{\rho^N(\mathsf{l}f)}{\rho^N(f)} \\ &= \frac{\rho(\mathsf{l}f) - \rho^N(\mathsf{l}f)}{\rho(\mathsf{l})} - \frac{\rho^N(\mathsf{l}f)(\rho(\mathsf{l}) - \rho^N(\mathsf{l}))}{\rho(\mathsf{l})\rho^N(\mathsf{l})} \\ &= \frac{\rho(\mathsf{l}f) - \rho^N(\mathsf{l}f)}{\rho(\mathsf{l})} - \frac{\pi^N(f)(\rho(\mathsf{l}) - \rho^N(\mathsf{l}))}{\rho(\mathsf{l})}. \end{aligned}$$

Using the basic inequality $(a-b)^2 \le 2(a^2+b^2)$, (4.7), that $|\pi^N(f)|^2 \le 1$ for all $|f|_\infty \le 1$ and that $|\mathsf{l}|_\infty|, |\mathsf{l}f|_\infty \le K$

$$\begin{aligned} \left| \mathbb{E}\Big[ \big(\pi(f) - \pi'(f)\big)^2 \Big] \right| &\le \frac{2}{\rho(\mathsf{l})^2} \left( \mathbb{E}\Big[ \big(\rho(\mathsf{l}f) - \rho^N(\mathsf{l}f)\big)^2 \Big] + \mathbb{E}\Big[ \big(\pi^N(f)\big)^2 \big(\rho(\mathsf{l}) - \rho^N(\mathsf{l})\big)^2 \Big] \right) \\ &\le \frac{4K^2}{\rho(\mathsf{l})^2} d(\rho, \rho^N)^2. \end{aligned}$$

Taking the supremum over test functions on the left-hand side and using that $\rho(\mathsf{l}) = Z$ gives us the desired result. □

We note that so far we did not specify how the samples $u^{(n)}$ defining the perturbed prior $\rho^N$ were generated. In fact, the result above holds for any empirical measure. If $u^{(n)} \sim \pi$ are sampled i.i.d., however, $\rho^N$ is a Monte Carlo approximation of $\rho$. Moreover, we can appeal to convergence results for Monte Carlo to show the convergence rate of $\pi^N$ to the true posterior $\pi$.

**Corollary 4.7.** *Let $\rho^N$ be a Monte Carlo estimator of $\rho$ in Theorem 4.6. Then, we have*

$$d(\pi, \pi^N) \leq \frac{2K}{Z} \frac{1}{\sqrt{N}}.$$

*Proof.* By Theorem 12.24, $d(\rho, \rho^N)^2 \leq \frac{1}{N}$. Using this in the right-hand side of (4.8) gives the desired result. □

## 4.4 Learning Regularizers for MAP Estimation

Again recall the inverse problem of finding $u \in \mathbb{R}^d$ from $y \in \mathbb{R}^k$ when related by (1.1):

$$y = G(u) + \eta.$$

We describe the idea of bilevel optimization to determine parameter $\theta \in \Theta \subseteq \mathbb{R}^p$ defining a regularizer in MAP estimation. The data assumption here differs from that employed in Section 4.1. We now assume (see Remark 1.3):

**Data Assumption 4.8.** *We are given samples $\{y^{(n)}, u^{(n)}\}_{n=1}^N$ assumed to be drawn i.i.d. from the joint probability distribution $\gamma$ on $(y, u) \in \mathbb{R}^k \times \mathbb{R}^d$ defined by equation (1.1) and Assumption 1.1.*

Such data is exactly what is required for the ideas of supervised learning from Chapter 14, where we use it to learn map from $u \in \mathbb{R}^d$ to $y \in \mathbb{R}$, though this is readily generalized to vector-valued output $y$. Note that, to solve the inverse problem, we would like to learn the inverse map from the data $y \in \mathbb{R}^k$ to the state $u \in \mathbb{R}^d$. We could try and do this directly using supervised learning. However, for inverse problems in which $k < d$ learning such a map may be difficult because of a lack of uniqueness; furthermore there is noise present in the $y^{(n)}$ which needs to be carefully accounted for. For these reasons it is arguably more informative to learn, from the supervised data defined by Data Assumption 4.8, how to regularize the inverse problem. This is the viewpoint we take here.

To understand the proposed methodology clearly it is important to distinguish between the one piece of data $y$ for which we wish to solve the inverse problem defined by (1.1) and the *training data pairs* $\{u^{(n)}, y^{(n)}\}_{n=1}^N$ (supervised data) which we assume are available to us, and which we use to learn the regularizer. This is different from the previous sections in this chapter where the training data from which we learn or define the prior was unsupervised data $\{u^{(n)}\}_{n=1}^N$.

We recall the posterior probability density function $\pi^y(u)$ on $u|y$ from Theorem 1.2, but assume that prior $\rho$ depends on an unknown parameter $\theta \in \Theta \subseteq \mathbb{R}^p$ so that the posterior has form

$$\pi^y(u; \theta) = \frac{1}{Z(\theta)} \nu(y - G(u)) \rho(u; \theta). \tag{4.9}$$

We will determine $\theta$ by choosing it so that MAP estimators for the inverse problem defined by data $y^{(n)}$ best match the paired data point $u^{(n)}$.

To this end recall the *loss function* $\mathsf{L}$ on data space $\mathbb{R}^k$ and the now $\theta-$dependent *regularizer* $\mathsf{R}$ defined by

$$\mathsf{L}(u;y) = -\log \nu\big(y - G(u)\big) = -\log \mathsf{l}(y|u), \quad \mathsf{R}(u;\theta) = -\log \rho(u;\theta), \tag{4.10}$$

leading to an *objective function* of the form

$$\mathsf{J}(u;y,\theta) = \mathsf{L}(u;y) + \mathsf{R}(u;\theta). \tag{4.11}$$

We have reverted to including the $y-$dependence in the likelihood. Note that we have also emphasized the parametric dependence of the loss and the objective function on $y \in \mathbb{R}^k$, something we did not do in Section 1.2. The MAP estimator can be viewed as a function, for each $\theta$, $u_{\mathrm{MAP}}(\cdot;\theta) : \mathbb{R}^k \to \mathbb{R}^d$:

$$u_{\mathrm{MAP}}(y;\theta) \in \arg\min_{u\in\mathbb{R}^d} \mathsf{J}(u;y,\theta).$$

We now introduce a loss function defined through a distance-like deterministic scoring rule $\mathsf{D} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$, as defined in Definition 12.56; the canonical example is the squared Euclidean norm. We then define the choice of $\theta$ through the optimization problem

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathbb{E}^{(y,u)\sim\gamma}\, \mathsf{D}\big(u, u_{\mathrm{MAP}}(y;\theta)\big).$$

This is referred to as bilevel optimization because of the optimization to find $u_{\mathrm{MAP}}$ which is used within the optimization to find $\theta^\star$. This procedure corresponds to hyperparameter tuning of $\theta$ on the validation set $\{y^{(n)}, u^{(n)}\}_{n=1}^N$. In practice the determination of $\theta^\star$ is implemented using $\gamma^N$ in place of $\gamma$ :

$$\theta^\star \in \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{n=1}^{N} \mathsf{D}\big(u^{(n)}, u_{\mathrm{MAP}}(y^{(n)};\theta)\big).$$

## 4.5 Learning Regularizers for Posterior Approximation

In this section we generalize the setting from the previous one to the more general problem of learning parameters in the prior to best approximate the posterior. We work under the following data assumption.

**Data Assumption 4.9.** *We are able to evaluate the likelihood* $\mathsf{l}(y|u)$ *for any pair* $(y,u) \in \mathbb{R}^k \times \mathbb{R}^d$. *We are given data in the form of samples* $\{y^{(n)}\}$ *from the marginal distribution* $\kappa$ *and samples* $\{u^{(n)}\}$ *from the prior distribution* $\rho$.

It is again helpful to recall the notation from Remark 1.3. The goal is to choose parameters $\theta$ in the prior so that the resulting posterior $\pi^y(u;\theta)$ in (4.9) is as close as possible to the true posterior $\pi^y(u)$ corresponding to the joint distribution $\gamma(y,u) = \mathbb{P}(y|u)\rho(u)$ underlying the data given in Data Assumption 4.9. We identify the parameters by minimizing the posterior error with respect to the KL divergence, in expectation over the data marginal on $y$ which, recall, has probability density function $\kappa$:

$$\mathsf{J}(\theta) := \mathbb{E}^y\Big[\mathsf{D}_{\mathrm{KL}}(\pi^y \| \pi^y(\cdot;\theta))\Big] = \int\Big[\int \log\Big(\frac{\pi^y(u)}{\pi^y(u;\theta)}\Big)\pi^y(u)du\Big]\kappa(y)\,dy. \tag{4.12}$$

Let

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{J}(\theta). \tag{4.13}$$

The following result shows that the optimal parameters can be computed without needing to evaluate the true posterior density.

**Theorem 4.10.** *The optimal parameter $\theta^\star$ in (4.13) corresponds to the solution of the optimization problem*

$$\theta^\star \in \arg\min_{\theta\in\Theta}\Big(\mathbb{E}^{u\sim\rho}[-\log\rho(u;\theta)] + \mathbb{E}^{y\sim\kappa}\Big[\log\int \nu(y-G(u'))\rho(u';\theta)\,du'\Big]\Big).$$

*Proof.* First note that $\gamma(y,u) = \pi^y(u)\kappa(y)$. The expected KL divergence in (4.12) can be decomposed into two terms as

$$\begin{aligned}\mathbb{E}^y\Big[\mathsf{D}_{\mathrm{KL}}(\pi^y\|\pi^y_\theta(\cdot;\theta))\Big] &= \int \gamma(y,u)\Big[\log\Big(\frac{\mathsf{l}(y|u)\rho(u)}{Z^y}\Big) - \log\Big(\frac{\mathsf{l}(y|u)\rho(u;\theta)}{Z^y(\theta)}\Big)\Big]\,dudy\\ &= \int \gamma(y,u)\Big[(\log\rho(u) - \log Z^y) - (\log\rho(u;\theta) - \log Z^y(\theta))\Big]\,dudy.\end{aligned} \tag{4.14}$$

Noticing that the first two terms are constant with respect to $\theta$, we only need to minimize with respect to the second term. That is,

$$\theta^\star \in \arg\min_{\theta\in\Theta}\int \gamma(y,u)[-\log\rho(u;\theta) + \log Z^y(\theta)]\,dudy.$$

Using the form of the normalizing constant $Z^y(\theta) = \int \nu(y-G(u'))\rho(u';\theta)\,du'$, we arrive at the objective above, after noticing that in the first term integration over $y$ is redundant, and in the second integration over $u$ is redundant. □

Remark 4.11. The expected KL divergence in (4.14) can also be written as

$$\begin{aligned}\mathbb{E}^y\Big[\mathsf{D}_{\mathrm{KL}}(\pi^y\|\pi^y_\theta(\cdot;\theta))\Big] &= \int \rho(u)[\log\rho(u) - \log\rho(u;\theta)]\,du + \int \kappa(y)\log\bigg(\frac{Z^y(\theta)}{Z^y}\bigg)\,dy.\\ &= \mathsf{D}_{\mathrm{KL}}(\rho\|\rho(\cdot;\theta)) - \mathsf{D}_{\mathrm{KL}}(\kappa\|\kappa(\cdot;\theta)),\end{aligned}$$

where in the last line we recognize that the normalizing constants $Z^y, Z^y(\theta)$ are equivalent to the marginal distributions of the observations $\kappa(y) = \int \nu(y-G(u'))\rho(u')\,du'$ and $\kappa(y;\theta) = \int \nu(y-G(u'))\rho(u';\theta)\,du'$, respectively. Thus, the objective in (4.12) involves two competing terms, the first relating only to the prior and the second only to the likelihood. ♢

## 4.6 Bibliography

The idea of learning prior probabilistic models from data is overviewed in [16]. An application is described in [244] where a generative adversarial network (GAN) is used to determine a mapping from a Gaussian to the space of prior data samples. The paper [103] also implicitly learns a prior, through simultaneous consideration of multiple inverse problems; but it also learns the posterior distribution for each of these inverse problems at the same time, linking to the two subsequent Chapters 5 and 6. Many methods of this type reduce the dimension of the unknown parameter space, determining a latent space of low effective dimension. A different approach to learning this mapping is to use invertible maps [172] and the work on normalizing flows [305, 293]; see [16] for application of invertible maps to prior construction for inversion. This idea can be combined with variational inference to solve sampling problems in general, and inverse problems in particular [270, 302, 101, 239]. The use of triangular transport maps for Bayesian inference was introduced in [88].

The idea of learning regularizers from data, to define objective functions for the optimization approach to inversion, is overviewed in [14] and [26]. The paper [292] provides a framework for the subject which employs structured factorizations of data matrices to learn semidefinite regularizers. The paper [204] develops an adversarial approach to the problem.

A collection of stability results relating the posterior error to perturbations in the prior measure with respect to various metrics and divergences (e.g., KL divergence, $\chi^2$ divergence and Wasserstein-1 metric) can be found in [104, 295]. We refer to Chapter 12 for background on these and other distances and divergences. While the stability results presented in this chapter are with respect to distance between two priors, the results can also be translated to stability with respect to the distance between the transport maps that define the prior. A set of these stability results for various metrics and divergences can be found in [22].

# Chapter 5

## Transporting to the Posterior

In this chapter we formulate various transport approaches to determining the posterior distribution, building on the material relating to transport in Chapter 13. In Section 5.1 we discuss mapping the prior to the posterior. Section 5.2 sets this work in the context of variational inference. In Section 5.3 we generalize and consider mapping general latent spaces, typically not the prior, to the posterior. Bibliographic remarks are contained in Section 5.4.

## 5.1 Learning the Prior to Posterior Map

Consider Bayes theorem written as map relating prior $\rho$ to posterior $\pi$ via the relation

$$\pi(u) = \frac{1}{Z}\mathsf{l}(u)\rho(u), \tag{5.1a}$$

$$Z = \mathbb{E}^{u\sim\rho}\big[\mathsf{l}(u)\big]. \tag{5.1b}$$

We drop explicit dependence on $y$ since we do not exploit it in this chapter. Equation (5.1) defines a map $\rho \mapsto \pi$ on the space of probability density functions. We now ask whether we can realize this map via an invertible transport map $T$ on $\mathbb{R}^d$ with the property $\pi = T_\sharp\rho$; equivalently, since $T$ is invertible, $\rho = (T^{-1})_\sharp\pi$. Specifically, following the approach in Section 13.2 as outlined in Remark 13.6, we seek to find $\theta \in \Theta \subseteq \mathbb{R}^p$ to minimize

$$\mathsf{F}(\theta) = \mathsf{D}_{\mathrm{KL}}(\rho\|T^{-1}(\cdot;\theta)_\sharp\pi). \tag{5.2}$$

From (13.4), we deduce that minimizing $\mathsf{F}(\cdot)$ over a class of diffeomorphisms (i.e., differentiable and invertible maps) $T$ is equivalent to minimizing $\mathsf{J}(\cdot)$ given by

$$\mathsf{J}(\theta) = -\mathbb{E}^{u\sim\rho}\Big[\log\pi\circ T(u;\theta) + \log\det D_uT(u;\theta)\Big].$$

Using the expression (5.1) for the posterior in terms of the prior, and noting that $Z$ is a constant with respect to $\theta$, we may instead write

$$\mathsf{J}(\theta) = -\mathbb{E}^{u\sim\rho}\Big[\log\rho\circ T(u;\theta) + \log\mathsf{l}\circ T(u;\theta) + \log\det D_uT(u;\theta)\Big]. \tag{5.3}$$

Finally, we set

$$\theta^\star \in \arg\min_{\theta\in\Theta}\mathsf{J}(\theta), \tag{5.4}$$

and employ transport map $T^\star = T(\cdot;\theta^\star)$. We observe that it is not necessary to know the normalization constant for the posterior in order to apply the methodology of this section.

Remark 5.1. We have sought the posterior as transport from the prior. This is natural in applications where the prior distribution is easy to sample from, and hence $\mathsf{J}(\theta)$ can be empirically approximated by Monte Carlo, for example. A key advantage of seeking transports, rather than directly learning an approximation to $\pi$ as in Chapter 2, is that it provides an easy approach to sample the posterior. In particular, if $z^{(n)} \sim \rho$ are i.i.d. reference samples, then $T^\star(z^{(n)}) \sim \pi$ are i.i.d. posterior samples. In this setting where $\rho$ corresponds to the prior density, we say that $T$ is a *prior-to-posterior transport map.* ♢

Remark 5.2. The terms in objective (5.3) are competitive. Minimizing the first two terms alone is achieved by setting $T(u;\theta) = u_{\text{MAP}}$ for all $u \in \mathbb{R}^d$: thus $T(u;\theta)$ maps to the MAP point of the posterior density, regardless of the input. On the other hand, the third and last term ensures the map remains strictly monotone, i.e., $D_u T(u;\theta) \succ 0$ for all $u \in \mathbb{R}^d$. This ensures that the map does not concentrate mass at a single point by returning a constant (and hence, non-invertible) function. The optimal solution is an invertible map taking prior samples $u \sim \rho$ to samples from the posterior. ♢

The following gives insight into how to parameterize the pushforward map $T$ if the posterior density is log-concave, by considering minimization of $\mathsf{J}(\cdot)$ over an infinite class of transports:

**Theorem 5.3.** *If the posterior density $\pi^y(u) \propto \rho(u)\mathsf{l}(u)$ is log-concave, then the optimization problem*

$$\arg\min_{T\in\mathcal{T}} -\mathbb{E}^{u\sim\rho}\Big[\log\rho\circ T(u) + \log \mathsf{l}\circ T(u) + \log\det D_u T(u)\Big]$$

*over the space of diffeomorphic, i.e., smooth and invertible, maps*

$$\mathcal{T} = \Big\{T \in C^1(\mathbb{R}^d;\mathbb{R}^d),\ \det D_u T(u) > 0 \textit{ for all } u \in \mathbb{R}^d\Big\}$$

*is convex.*

*Proof.* If $\pi^y(u)$ is log-concave, then $-\log\big(\rho(u)\mathsf{l}(u)\big) = -\log\rho(u) - \log\mathsf{l}(u)$ is convex. By the convexity of $u \mapsto -\log\rho(u) - \log\mathsf{l}(u)$ and $u \mapsto -\log\det(u)$ we have that $T \mapsto \mathbb{E}^{u\sim\rho}\big[-\log\rho\big(T(u)\big) - \log\mathsf{l}\big(T(u)\big) - \log\det D_u T(u)\big]$ is convex.

Given that $tT_1 + (1-t)T_2 \in \mathcal{T}$ for all $T_1, T_2 \in \mathcal{T}$ and $t \in [0,1]$, then $\mathcal{T}$ is a convex set. Then minimizing a convex functional over a convex set yields the result. □

Remark 5.4. This theorem demonstrates that, if the posterior is log-concave, then parameterization of $\theta \mapsto T(u;\theta)$ restricted to $\mathcal{T}$ will lead to a desirable objective function for the purpose of optimization. If $\pi^y$ is not strongly log-concave, or the map parameterization $\theta \mapsto T(u;\theta)$ is not convex for all $u \in \mathbb{R}^d$, then the optimization problem of minimizing (5.3) is in general non-convex. ♢

## 5.2 Connection to Variational Inference

In this section we consider parameterized transports. Let

$$\mathcal{T} := \Big\{ T(\cdot;\theta) \in C^1(\mathbb{R}^d;\mathbb{R}^d), \theta \in \Theta \,\Big|\, \det D_u T(u;\theta) > 0 \text{ for all } (u,\theta) \in \mathbb{R}^d \times \Theta \Big\}$$

be a space of smooth and invertible maps depending on parameters $\theta$.

**Theorem 5.5.** *Let $\mathcal{Q} := \{q : q = T_\sharp \rho, T \in \mathcal{T}\}$. Then, the optimal parameters $\theta^\star$ solving (5.4), with $\mathsf{J}(\cdot)$ defined by (5.3), also solve the variational inference problem*

$$q^\star \in \arg\min_{q \in \mathcal{Q}} \mathsf{D}_{\mathrm{KL}}(q\|\pi),$$

*by setting $q^\star := T(\cdot;\theta^\star)_\sharp \rho$.*

*Proof.* We showed above that solving the optimization problem (5.4), with $\mathsf{J}(\cdot)$ defined by (5.3), is equivalent to minimizing $\mathsf{F}(\cdot)$ from (5.2). But, from Theorem 12.34 on the invariance of the KL divergence under invertible and differentiable transformations, we have that

$$\mathsf{D}_{\mathrm{KL}}(\rho\|T^{-1}(\cdot,\theta)_\sharp \pi) = \mathsf{D}_{\mathrm{KL}}(T(\cdot;\theta)_\sharp \rho\|\pi).$$

Thus

$$\arg\min_{\theta \in \Theta} \mathsf{J}(\theta) = \arg\min_{\theta \in \Theta} \mathsf{D}_{\mathrm{KL}}(T(\cdot;\theta)_\sharp \rho\|\pi).$$

Lastly, $T(\cdot,\theta^\star) : \mathbb{R}^d \to \mathbb{R}^d$ is a differentiable map, and its derivative has positive determinant; it is thus an invertible map. Thus $T(u;\theta^\star)_\sharp \rho \in \mathcal{Q}$. □

**Example 5.6.** Consider the setting where the prior is a standard Gaussian $\mathcal{N}(0,I)$. Now assume that that $T$ is affine: $T(u) = Au + m$, for a positive definite matrix $D_u T(u) = A \in \mathbb{R}^{d\times d}_{\geq 0}$ and vector $m \in \mathbb{R}^d$. An affine transformation of a standard Gaussian yields a Gaussian random variable. In fact, the class of approximate posterior distributions then consists of multivariate Gaussians of the form

$$\mathcal{Q} = \Big\{ q = \mathcal{N}(m, AA^\top),\ m \in \mathbb{R}^d, A \in \mathbb{R}^{d\times d}_{\geq 0} \Big\}.$$

Hence, solving the optimization problem for $\mathsf{J}(\cdot)$ is equivalent to the Gaussian variational inference problem considered in Chapter 2, parameterizing the Gaussian covariance through a square root; for example it is possible to parameterize the covariance using its Cholesky factorization by considering triangular matrices $A \in \mathbb{R}^{d\times d}$ in $\mathcal{Q}$ with positive diagonal entries. ♢

**Example 5.7.** Let $\rho$ be a probability density function in the mean-field family in Definition 2.5. Furthermore let $T$ be a diagonal map of the form $T(u) = (T_1(u_1), \dots, T_d(u_d))$ where $T_i \colon \mathbb{R} \to \mathbb{R}$ are univariate differentiable and invertible transformations: that is $D_{u_i} T_i(u_i) > 0$ for all $u_i \in \mathbb{R}$. Then, the class of approximate posterior distributions corresponds to

$$\mathcal{Q} = \Bigg\{ q(u) = \prod_i q_i(u_i),\ q_i = (T_i)_\sharp \rho_i \Bigg\}.$$

♢

## 5.3 Learning Other Posterior Maps

So far we looked for a map pushing forward the prior to the posterior. This is advantageous when we would like the posterior to inherit certain properties of the prior. For example, if the prior and posterior have the same tail behavior, then the transport map $T$ only needs to depart from an identity map in the bulk of the distribution. Similarly, if the posterior and prior only differ on a low-dimensional subspace of the parameters $u \in \mathbb{R}^d$, then the map $T$ can be represented using a *ridge function* that is constant for inputs orthogonal to the subspace of interest.

In some settings, it may be computationally expedient to seek $\pi$ which is the pushforward of a different, easy-to-sample, reference density $\varrho(u)$ on $\mathbb{R}^d$, (such as a multivariate Gaussian) rather than the prior $\rho(u)$ (when it is not Gaussian). This is particularly convenient when it is challenging to sample from the prior density. In these cases we may minimize a similar objective to (5.3) that changes only in the definition of the measure over which we take the expectation. That is, we may learn the parameters $\theta \in \Theta$ of the map $T$ by minimizing

$$\mathsf{J}(\theta) = -\mathbb{E}^{u\sim\varrho}\Big[\log\rho\circ T(u;\theta) + \log\mathsf{l}\circ T(u;\theta) + \log\det D_u T(u;\theta)\Big]. \tag{5.5}$$

Letting $\theta^\star$ denote the optimal parameters, the resulting posterior approximation is given by $\pi^y \approx T(\cdot,\theta^\star)_\sharp\varrho$.

Remark 5.8. Unlike the variational inference problem in Chapter 2, we now have two degrees of freedom: the reference measure and the map. In principle, one may even parameterize the reference measure $\varrho(\cdot,\vartheta)$ (e.g., as a mixture of Gaussians with unknown means, covariances, and mixture weights) and learn its parameters $\vartheta$ simultaneously to learning the map $T(\cdot,\theta)$. ♢

## 5.4 Bibliography

The approach to seek prior-to-posterior maps was proposed in [88]. In particular that paper parameterized $T$ as a monotone-triangular map known as the Knothe-Rosenblatt rearrangement; the advantage of this choice of map is that the determinant Jacobian, required to define the minimization, is easy-to-evaluate. The approach was extended to compositions of triangular maps, commonly referred to as normalizing flows, in [270]. An approach to seek prior to posterior maps that are optimal with respect to minimizing a transportation distance (see Subsection 12.1.3 for a discussion on optimal transport) was proposed in [306].

An alternative to seeking the transport map is to construct the map incrementally by learning a map that is a small perturbation of the identity that minimizes the distance between the current approximation and the actual posterior distribution. Stein-Variational Gradient Descent is one such algorithm; it proceeds by seeking the map in a reproducing kernel Hilbert space in which the distance is measured by the KL divergence [199].

To handle certain classes of target distributions, it is sometimes desirable to consider objectives other than the KL divergence; examples include use of $\alpha$-divergences [139], or the development of tailored transport map approximations for heavy-tailed posteriors [198].

# Chapter 6

## Learning Dependence on Data

Again our focus is on the inverse problem of finding $\pi^y(u)$ given by

$$\pi^y(u) = \frac{1}{Z}\mathsf{l}(y|u)\rho(u), \tag{6.1a}$$

$$Z = \mathbb{E}^{u\sim\rho}\big[\mathsf{l}(y|u)\big]. \tag{6.1b}$$

In this chapter, unlike the previous one, dependence of the target measure on $y$ is important, and so we retain it explicitly in the notation $\pi^y$. Employing the notation from Remark 1.3, we describe the focus of this chapter. We assume that $\pi^y(u)$ is defined by conditioning the joint distribution $\gamma(y,u)$ on a specific realization $y$; we recall that $\kappa(y) = \int \gamma(y,u)\,du$ denotes the marginal distribution for the observations. Our goal, then, is to learn a map depending on the observation $y$ that can be used to sample from the posterior corresponding to *any* realization of $y$. The resulting model can then be repeatedly used for different realizations, rather than having to re-learn a different map for each $y$. This reduces the cost of multiple inference procedures to training a single model. Hence it is known as *amortized inference.* We will introduce approaches to amortized inference where the likelihood $\mathsf{l}(y|u)$ can be evaluated during training in Section 6.1; and we will introduce likelihood-free approaches in Section 6.2. In Section 6.2 we discuss the consequence of having block-triangular pushforwards; and we discuss the learning of block-triangular transport maps, encoding data dependence in a natural way. Section 6.3 discusses the learning of likelihoods. Our methodology sections conclude with a brief discussion of learning observation data dependence on MAP estimators in Section 6.4. We end with Section 6.5 containing bibliographic remarks.

## 6.1 Likelihood-Based Inference

In this section we generalize variational inference to learn the dependence of the optimal approximation on the data $y$. Our assumptions about the set-up in this section are as follows:

**Data Assumption 6.1.** *We are able to evaluate the likelihood* $\mathsf{l}(y|u)$ *for any pair* $(y,u) \in \mathbb{R}^k \times \mathbb{R}^d$. *We are given data in the form of independent samples* $\{y^{(n)}\}_{n=1}^N$ *from the marginal distribution* $\kappa$ *and independent samples* $\{u^{(n)}\}_{n=1}^N$ *from the prior distribution* $\rho$.

Remark 6.2. In the preceding Data Assumption 6.1 the set of independent samples $\{y^{(n)}\}_{n=1}^N$ and the set of independent samples $\{u^{(n)}\}_{n=1}^N$ do not need to be independent of one another, and the same number of each is not required. However, in practice they are often found by marginalizing the data set of samples $\{(y^{(n)}, u^{(n)})\}_{n=1}^N$ from the joint distribution $\gamma$ and then it is natural to have the same number of samples of each. $\diamondsuit$

With the goal of generalizing variational inference to learn the dependence of the optimal approximation on the data, we define the optimization problem

$$q^\star \in \arg\min_{q\in\mathcal{Q}} \mathbb{E}^{y\sim\kappa}\big[\mathsf{D}_{\mathrm{KL}}(q(\cdot;y)\|\pi^y)\big].$$

As in standard variational inference from Chapter 2, $\mathcal{Q}$ may for instance be the mean-field family or a parameterized family of probability density functions on $\mathbb{R}^d$. In the latter case, the parameters may be themselves be parameterized to reflect dependence on the observations $y \in \mathbb{R}^k$.

**Example 6.3.** This generalized variational inference problem can be implemented by seeking a mean-field approximation or a Gaussian approximation where the parameters are functions of the observations. For instance, $\mathcal{Q}$ may contain multivariate Gaussian densities $q(\cdot;y) = \mathcal{N}(m(y), \Sigma(y))$ where the mean $m(y)$ and covariance $\Sigma(y)$ depend on the observation. This dependence on $y$ will itself need to be parameterized, for example as a linear function, or using neural networks, random features or Gaussian processes (see Chapter 14) constrained to ensure the covariance is a positive semi-definite matrix for all $y$. $\diamondsuit$

Generalizing the transport approach to variational inference described in Chapter 5, we can define the family of approximating distributions by using a transport map that pushes forward a simple reference distribution (for instance the prior $\rho$) and is also parameterized by $y$. In this case, we seek a transport map $T\colon \mathbb{R}^d \times \mathbb{R}^k \times \mathbb{R}^p \to \mathbb{R}^d$ that depends on both input parameters and observations so that $u \mapsto T(u;y,\theta)$ defines an invertible transport map approximating the pushforward of $\rho$ to the posterior $\pi^y(u)$, for each choice of the observations $y$. We seek this transport as the minimizer of the following objective

$$\begin{aligned}
\mathsf{F}(\theta) &= \mathbb{E}^{y\sim\kappa}\big[\mathsf{D}_{\mathrm{KL}}(T(\cdot;y,\theta)_\sharp\rho\|\pi^y)\big] && (6.3a)\\
&= \mathbb{E}^{y\sim\kappa}\big[\mathsf{D}_{\mathrm{KL}}(\rho\|T^{-1}(\cdot;y,\theta)_\sharp\pi^y)\big], && (6.3b)\\
\theta^\star &\in \arg\min_{\theta\in\Theta} \mathsf{F}(\theta), && (6.3c)
\end{aligned}$$

where $\Theta \subseteq \mathbb{R}^p$. (In going from (6.3a) to (6.3b) we have used the invariance of the KL divergence under invertible transformations, Theorem 12.34). Following the approach in Chapter 5, we can rewrite this objective for the parameters via minimization of a loss that may be approximated empirically: [1]

$$\mathsf{J}(\theta) = -\mathbb{E}^{(y,u)\sim\kappa\otimes\rho}\Big[\log\rho(T(u;y,\theta)) + \log \mathsf{l}(y|T(u;y,\theta)) + \log\det D_u T(u;y,\theta)\Big].$$

[1] This is the analog of equation (5.3), the objective function in the case of a fixed single instance of data $y$.

## 6.2 Likelihood-Free Inference

A common setting arising in many inverse problems is one in which the likelihood $\mathsf{l}(y|u)$ is not analytically available or tractable to evaluate. We work in this section under the following assumption, which should be compared with Assumption 6.1.

**Data Assumption 6.4.** *We are able to sample from the likelihood $\mathsf{l}(\cdot|u)$ for any $u \in \mathbb{R}^d$. We are given data in the form of independent samples $\{(y^{(n)}, u^{(n)})\}_{n=1}^N$ from the joint distribution $\gamma$.*

Thus here we show how to construct posterior approximations without requiring evaluations of $\mathsf{l}(\cdot|\cdot)$, unlike in Section 6.1. Instead, we rely on sampling the joint distribution $\gamma(y, u)$ to learn the dependence between states and observations. Sampling $\gamma$ is feasible by sampling $u$ from the prior $\rho(u)$ and then sampling a synthetic observation $y$ from the likelihood model conditioned on $u$. Hence, this is known as *likelihood-free* or *simulation-based* inference. Note, in particular, that we are assuming that sampling from the likelihood is straightforward, even though evaluation of it is not.

Remark 6.5. The likelihood function may require marginalizing with respect to a latent random variable $z$. That is,

$$\mathsf{l}(y|u) = \int_{\mathbb{R}^{d_z}} \mathbb{P}(y|u, z)\mathbb{P}(z|u)\, dz. \tag{6.4}$$

When $d_z \gg 1$, it will be difficult to evaluate the integration over $z$. But we can simulate from the joint distribution $\mathbb{P}(y, u) = \mathsf{l}(y|u)\mathbb{P}(u)$, with likelihood as in (6.4), as follows. First sample $u$ from the prior $\rho$; then sample latent state $z$ from $\mathbb{P}(z|u)$; and finally sample $y$ from $\mathbb{P}(y|u, z)$. The collection $(y, z, u)$ is a sample from the distribution $\mathbb{P}(y, z, u)$. The subset of pairs $(y, u)$ are then samples from the distribution $\mathsf{l}(y|u)\rho(u)$. ♢

**Example 6.6.** A likelihood with high-dimensional latent variables arises when performing parameter inference in a *hidden Markov model* given noisy observations of the hidden state. That is, let $z = (z_1, \dots, z_J)$ and $y = (y_1, \dots, y_J)$ be the states and observations of a dynamical system, respectively, with $z_j \in \mathbb{R}^{dz}$ and $y_j \in \mathbb{R}^k$. The states and observations follow the dynamics and observation models

$$\begin{aligned} z_{j+1} &= \Psi(z_j; u) + \xi_j, \quad \xi_j \sim \mathcal{N}(0, \Sigma), \\ y_{j+1} &= h(z_{j+1}) + \eta_{j+1}, \quad \eta_{j+1} \sim \mathcal{N}(0, \Gamma), \end{aligned}$$

for $j = 0, \dots, J-1$ where $\Psi\colon \mathbb{R}^{d_z} \times \mathbb{R}^d \to \mathbb{R}^{d_z}$ is a nonlinear map depending on the parameters $u$, the initial condition $z_0 \in \mathbb{R}^{d_z}$ is known, and $h\colon \mathbb{R}^d \to \mathbb{R}^k$. The conditional probabilities in (6.4) have the closed forms

$$\begin{aligned} \mathbb{P}(y|u, z) &= \prod_{j=1}^{J} \mathcal{N}(y_j; h(z_j), \Gamma), \\ \mathbb{P}(z|u) &= \prod_{j=0}^{J-1} \mathcal{N}(z_{j+1}; \Psi(z_j; u), \Sigma). \end{aligned}$$

However, the likelihood for the marginal variables is not available in closed-form due to the possible non-linearity in $\Psi$ and $h$. ♢

### 6.2.1 Consequences of Block-Triangular Pushforward

In this setting the approach we introduce is to construct a map to approximate the joint distribution $\gamma(y,u)$ in such a way that we can extract the $y-$parameterized family of conditionals $\pi^y(u)$. To this end, we make the following assumption:

**Assumption 6.7.** *Let $\varrho(y,u) = \varrho_1(y)\varrho_2(u)$ be a product reference density on $\mathbb{R}^k \times \mathbb{R}^d$ and $T\colon \mathbb{R}^k \times \mathbb{R}^d \to \mathbb{R}^k \times \mathbb{R}^d$ be a transport map with the property that the joint density $\gamma(y,u) = \kappa(y)\pi^y(u)$ is a pushforward under $T$, i.e., $\gamma = T_\sharp \varrho$. Assume also that $T$ is* block-triangular*:* [2]

$$T(y,u) = \begin{bmatrix} T_1(y) \\ T_2(y,u) \end{bmatrix}, \tag{6.5}$$

*where $T_1 : \mathbb{R}^k \to \mathbb{R}^k$ is invertible and $T_2(y,\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ is invertible for every $y \in \mathbb{R}^k$.*

Remark 6.8. We discuss training based on the pushforward of a block-triangular map in Subsection 6.2.2, a context in which $T^{-1}$ arises. To avoid inverting the map during training, it is often convenient to directly work with the inverse map $S \coloneqq T^{-1}$. For a block-triangular $T$, the inverse $S$ is also a block-triangular map of the form

$$S(y,u) = \begin{bmatrix} S_1(y) \\ S_2(y,u) \end{bmatrix}, \tag{6.6}$$

where $S_1 : \mathbb{R}^k \to \mathbb{R}^k$ is the inverse of $T_1$ and $S_2(y,\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ is the inverse of $T_2(T_1^{-1}(y),\cdot)$ for each $y \in \mathbb{R}^k$. By writing $T_\sharp\varrho = (S^{-1})_\sharp\varrho$, the push-forward density can be more easily evaluated in terms of the map $S$, rather than the inverse of $T$. As well as being useful computationally, these definitions of $S_1, S_2$ simplify various expressions in the proof of the following theorem. $\diamondsuit$

The next theorem motivates the choice of tensor-product reference measure: such a choice ensures that the block-triangular form of $T$ provides a map that can be used to readily characterize the posterior density.

**Theorem 6.9.** *Let Assumption 6.7 hold. Then $(T_1)_\sharp\varrho_1 = \kappa$ and*

$$T_2(T_1^{-1}(y),\cdot)_\sharp\varrho_2 = \pi^y. \tag{6.7}$$

Remark 6.10. The theorem shows that, if $w \sim \varrho_2$ then $T_2\big(T_1^{-1}(y), w\big) \sim \pi^y$. Choosing $\varrho_2$ to be an easy distribution to sample, and learning $T$ from data, then leads to a method for sampling from the posterior. In other words, imposing block-triangular structure on the map and the product form for the reference measure yields a map that meets our goal of characterizing the posterior distribution. $\diamondsuit$

[2] The map is called block-triangular because, if the map is differentiable, then its Jacobian is given by a block-triangular matrix.

*Proof of Theorem 6.9.* We first prove that $(T_1)_\sharp \varrho_1 = \kappa$. Let $S = T^{-1}$ be the inverse map of the form in (6.6). By Lemma 12.33 we have that

$$T_\sharp \varrho(y,u) = \varrho_1\big(S_1(y)\big) \det DS_1(y) \varrho_2\big(S_2(y,u)\big) \det D_u S_2(y,u), \tag{6.8a}$$

$$(T_1)_\sharp \varrho_1(y) = \varrho_1\big(S_1(y)\big) \det DS_1(y), \tag{6.8b}$$

$$\big(T_2(T_1^{-1}(y),\cdot)\big)_\sharp \varrho_2(u) = \varrho_2\big(S_2(y,u)\big) \det D_u S_2(y,u). \tag{6.8c}$$

Integrating the last identity, using that the left-hand side is a probability density function, gives for all $y \in \mathbb{R}^k$,

$$1 = \int_{\mathbb{R}^d} \varrho_2\big(S_2(y,u)\big) \det D_u S_2(y,u)\, du.$$

Hence, multiplying identity (6.8a) by an arbitrary test function $\psi : \mathbb{R}^k \to \mathbb{R}$ and integrating over $y$ and $u$ we obtain

$$\int_{\mathbb{R}^k} \int_{\mathbb{R}^d} \psi(y) T_\sharp \varrho(y,u)\, dydu = \int_{\mathbb{R}^k} \psi(y) \varrho_1\big(S_1(y)\big) \det DS_1(y)\, dy. \tag{6.9}$$

But $T_\sharp \varrho(y,u) = \gamma(y,u) = \kappa(y)\pi^y(u)$. Thus, using the fact that, for all $y \in \mathbb{R}^k$,

$$1 = \int_{\mathbb{R}^d} \pi^y(u)\, du$$

we see that (6.9) simplifies to give

$$\int_{\mathbb{R}^k} \psi(y)\kappa(y)\, dy = \int_{\mathbb{R}^k} \psi(y) \varrho_1\big(S_1(y)\big) \det DS_1(y)\, dy. \tag{6.10}$$

Since this is true for all $\psi$ we deduce that $\kappa(y) = \varrho_1\big(S_1(y)\big) \det DS_1(y)$ and this, by Lemma 12.33, is the desired result.

We now prove identity (6.7). Recall from Remark 6.8 that

$$S_2(y,\cdot) = T_2^{-1}(T_1^{-1}(y),\cdot)$$

for each fixed $y$. Multiplying and dividing the right-hand side of (6.8c) by the right-hand side of (6.8b), we have

$$\begin{aligned} T_2(T_1^{-1}(y),\cdot)_\sharp \varrho_2(u) &= \varrho_2\big(S_2(y,u)\big) \det D_u S_2(y,u) \\ &= \frac{\varrho_1\big(S_1(y)\big)\varrho_2\big(S_2(y,u)\big)}{\varrho_1\big(S_1(y)\big)} \frac{\det DS_1(y)}{\det DS_1(y)} \det D_u S_2(y,u) \\ &= \frac{\varrho\big(S(y,u)\big) \det D_{(y,u)} S(y,u)}{\varrho_1\big(S_1(y)\big) \det DS_1(y)}. \end{aligned} \tag{6.11}$$

In the last equality we used the product form of the reference probability density function and we used the fact that the determinant of a block-triangular matrix can be written as a product of the determinant of its diagonal blocks.

Now, note that the numerator of (6.11) is the joint density $T_\sharp \varrho(y,u) = \gamma(y,u)$ and the denominator is the marginal $\kappa(y)$ on $y$ under $\gamma$, as shown in (6.10). Thus, by conditioning the joint density $\gamma(y,u)$ on $y$ we find that

$$\frac{\varrho(S(y,u)) \det D_{(y,u)} S(y,u)}{\varrho_1(S_1(y)) \det DS_1(y)} = \frac{\gamma(y,u)}{\kappa(y)} = \pi^y(u).$$

Hence we have shown the desired result that

$$T_2(T_1^{-1}(y), \cdot)_\sharp \varrho_2(u) = \frac{\gamma(y,u)}{\kappa(y)} = \pi^y(u).$$

□

Remark 6.11. While a block-triangular map can sample from conditional distributions, it does not uniquely determine a particular map with the desired property. Indeed, when such a map exists, there may exist an infinite number of maps with the structure in Assumption 6.7. We refer to Section 13.2 for discussion of computational methods to find transports from data. ♢

Remark 6.12. The reference measure $\varrho$ is a degree of freedom in the measure transport framework. A useful choice is $\varrho(y,u) = \kappa(y)\varrho_2(u)$; that is, to choose $\varrho_1(y) = \kappa(y)$. It then follows that the identity map $T_1(y) = y$ is a valid transport, trivially pushing forward $\varrho_1(y) = \kappa(y)$ to $\kappa(y)$. This choice avoids the inversion of $T_1$ in (6.7) and makes sampling from $\pi_1$ straightforward: once we have determined $T_2$ compatible with this choice of $T_1$ then for all $y \sim \kappa$ we have

$$T_2(y,u) \sim \pi^y, \quad \text{for} \quad u \sim \varrho_2.$$

♢

**Example 6.13.** Let $\varrho(y,u)$ be the standard Gaussian distribution on $\mathbb{R}^k \times \mathbb{R}^d$ and let $\gamma(y,u)$ be the Gaussian distribution $\mathcal{N}(m,\Sigma)$ on $\mathbb{R}^k \times \mathbb{R}^d$ with mean and covariance

$$m = \begin{bmatrix} m_y \\ m_u \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} \Sigma_{yy} & \Sigma_{yu} \\ \Sigma_{uy} & \Sigma_{uu} \end{bmatrix},$$

where $\Sigma_{uy} = \Sigma_{yu}^\top$. Let $T\colon \mathbb{R}^k \times \mathbb{R}^d \to \mathbb{R}^k \times \mathbb{R}^d$ be a block-triangular transport map of the form

$$T(y,u) = \begin{bmatrix} T_1(y) \\ T_2(y,u) \end{bmatrix} = \begin{bmatrix} m_y + \Sigma_{yy}^{1/2} y \\ m_{u|y}(m_y + \Sigma_{yy}^{1/2} y) + \Sigma_{u|y}^{1/2} u \end{bmatrix},$$

where $m_{u|y}(y) \coloneqq m_u + \Sigma_{uy}\Sigma_{yy}^{-1}(y - m_y)$ and $\Sigma_{u|y} \coloneqq \Sigma_{uu} - \Sigma_{uy}\Sigma_{yy}^{-1}\Sigma_{yu}$ denote the conditional mean and covariance matrix of $u$ given $y$. Note that

$$T(y,u) = \begin{bmatrix} m_y + \Sigma_{yy}^{1/2} y \\ m_u + \Sigma_{uy}\Sigma_{yy}^{-1/2} y + \Sigma_{u|y}^{1/2} u \end{bmatrix}.$$

For $(y, u)$ distributed according to the standard Gaussian distribution on $\mathbb{R}^k \times \mathbb{R}^d$ straightforward calculation shows that $T(y, u)$ has distribution $\mathcal{N}(m, \Sigma)$ on $\mathbb{R}^k \times \mathbb{R}^d$.

Alternatively we may show that $T_\sharp \varrho = \gamma$ by working with densities. To this end we will compute the pushforward density $T_\sharp \varrho(y, u) = \varrho(T^{-1}(y, u)) \det DT^{-1}(y, u)$. First, the inverse map has the form

$$T^{-1}(y,u) = \begin{bmatrix} T_1^{-1}(y) \\ T_2^{-1}(y,u) \end{bmatrix} = \begin{bmatrix} \Sigma_{yy}^{-1/2}(y - m_y) \\ \Sigma_{u|y}^{-1/2}(u - m_{u|y}(y)) \end{bmatrix}.$$

Then, for the standard Gaussian reference $\varrho$ we have

$$\begin{aligned} \varrho(T^{-1}(y,u)) &= \frac{1}{\sqrt{(2\pi)^{k+d}}} \exp\Big(-\frac{1}{2}|\Sigma_{yy}^{-1/2}(y - m_y)|^2 - \frac{1}{2}|\Sigma_{u|y}^{-1/2}(u - m_{u|y}(y))|^2\Big) \\ &= \frac{1}{\sqrt{(2\pi)^{k+d}}} \exp\left(-\frac{1}{2}\left|\begin{bmatrix} \Sigma_{yy}^{-1/2} & 0 \\ -\Sigma_{u|y}^{-1/2}\Sigma_{uy}\Sigma_{yy}^{-1} & \Sigma_{u|y}^{-1/2} \end{bmatrix}\begin{bmatrix} y - m_y \\ u - m_u \end{bmatrix}\right|^2\right) \\ &= \frac{1}{\sqrt{(2\pi)^{k+d}}} \exp\left(-\frac{1}{2}\left|\Sigma^{-1/2}\begin{bmatrix} y - m_y \\ u - m_u \end{bmatrix}\right|^2\right), \end{aligned}$$

where $\Sigma^{-1/2}$ denotes the inverse of the block Cholesky factor of $\Sigma$. We note that the block-Cholesky factor $\Sigma^{1/2}$ satisfies

$$\Sigma^{1/2}(\Sigma^{1/2})^\top = \begin{bmatrix} \Sigma_{yy}^{1/2} & 0 \\ \Sigma_{uy}\Sigma_{yy}^{-1/2} & \Sigma_{u|y}^{1/2} \end{bmatrix}\begin{bmatrix} \Sigma_{yy}^{1/2} & \Sigma_{yy}^{-1/2}\Sigma_{yu} \\ 0 & \Sigma_{u|y}^{1/2} \end{bmatrix} = \begin{bmatrix} \Sigma_{yy} & \Sigma_{yu} \\ \Sigma_{uy} & \Sigma_{uu} \end{bmatrix} = \Sigma.$$

Moreover, the determinant of the block-triangular Jacobian of $T^{-1}$ is given by the product of the determinant of the Jacobian's diagonal elements. That is,

$$\det DT^{-1}(y,u) = \det \Sigma_{yy}^{-1/2} \det \Sigma_{u|y}^{-1/2} = (\det \Sigma_{yy})^{-1/2}(\det \Sigma_{u|y})^{-1/2} = (\det \Sigma)^{-1/2},$$

where in the last equality we used the formula for the determinant of a block matrix with invertible diagonal elements.

Thus, the pushforward density is given by

$$\varrho(T^{-1}(y,u)) \det DT^{-1}(y,u) = \frac{1}{\sqrt{(2\pi)^{k+d}\det\Sigma}} \exp\left(-\frac{1}{2}\begin{bmatrix} y - m_y \\ u - m_u \end{bmatrix}^\top \Sigma^{-1}\begin{bmatrix} y - m_y \\ u - m_u \end{bmatrix}\right).$$

That is, the pushforward is a multivariate Gaussian with mean $m$ and covariance $\Sigma$, which matches the density for $\gamma$. Moreover, $T_1(y) = m_y + \Sigma_{yy}^{1/2} y$ pushes forward $\varrho_1 = \mathcal{N}(0, I_k)$ to $\kappa = \mathcal{N}(m_y, \Sigma_{yy})$, and so by Theorem 6.9 we have that

$$T_2(T_1^{-1}(y), u) = m_{u|y}(y) + \Sigma_{u|y}^{1/2} u$$

pushes forward $\mathcal{N}(0, I_d)$ to the posterior $\pi^y$ for each $y$. $\diamondsuit$

### 6.2.2 Learning Block-Triangular Pushforward Maps

Once again, recall the notation of Remark 1.3. In this subsection we discuss the learning of block-triangular transport maps, given data pairs $\{(y^{(n)}, u^{(n)})\}_{n=1}^N$ from the joint distribution $\gamma(y,u)$. Let $\theta \in \Theta \subseteq \mathbb{R}^p$ denote the parameters of the block-triangular map $T(u,y;\theta)$ given by (6.5). Our goal is to find the optimal parameters by minimizing the KL divergence between the pushforward distribution $T_\sharp \varrho$ and the joint distribution $\gamma$. That is,

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{D}_{\mathrm{KL}}(\gamma \| T(\cdot;\theta)_\sharp \varrho). \tag{6.12}$$

Remark 6.14. Compare with the minimization problems (5.2) and (6.3) for which the reference measure appears in the left-hand argument of $\mathsf{D}_{\mathrm{KL}}(\cdot\|\cdot)$. This choice is dictated by the form of the data, drawn from the joint distribution; and specifically drawn by sampling from prior on $u$ and then from likelihood on $y|u$. The following theorem shows that the optimal parameters can then be identified from an optimization objective that only depends on the joint distribution $\gamma(y,u)$ through an expectation, and hence can be solved using an empirical approximation of the objective. ♢

**Theorem 6.15.** *Let $T$ be a block-triangular transport map of the form in Assumption 6.7, let $\varrho(y,u) = \kappa(y)\varrho_2(u)$ be a reference distribution as in Remark 6.12 with $T_1(y) = y$ and let $\theta^\star$ be defined by (6.12). Then, the optimal parameters for the map $T_2$ are given by*

$$\theta^\star \in \arg\min_{\theta\in\Theta} -\mathbb{E}^{(y,u)\sim\gamma}\Big[\log \varrho_2 \circ T_2^{-1}(y,u;\theta) + \log\det D_u T_2^{-1}(y,u;\theta)\Big],$$

*where $T_2^{-1}(y,u;\theta)$ denotes the inverse of the function $u \mapsto T_2(y,u;\theta)$ for each given pair $(y,\theta)$.*

*Proof.* From the chain rule for the KL divergence, Lemma 12.29, we have

$$\mathsf{D}_{\mathrm{KL}}(\gamma\|T(\cdot;\theta)_\sharp\varrho) = \mathsf{D}_{\mathrm{KL}}(\kappa\|(T_1)_\sharp\kappa) + \mathbb{E}^{y\sim\kappa}[\mathsf{D}_{\mathrm{KL}}(\pi^y(u)\|T_2(y,\cdot;\theta)_\sharp\varrho_2)]. \tag{6.13}$$

Setting $T_1(y) = y$, the first term in (6.13) is zero, and the second term is given by

$$\begin{aligned}\mathbb{E}^{y\sim\kappa}[\mathsf{D}_{\mathrm{KL}}(\pi^y\|T_2(y,\cdot;\theta)_\sharp\varrho_2)] &= \int\int \kappa(y)\pi^y(u)\Big[\log\pi^y(u) - \log(T_2(y,\cdot;\theta)_\sharp\varrho_2(u))\Big]\,dydu \\ &= c - \int\int \gamma(y,u)\log(T_2(y,\cdot;\theta)_\sharp\varrho_2(u))\ dudy,\end{aligned}$$

where $c$ is a constant that is independent of the parameters $\theta$. Therefore, minimizing the second term achieves the minimum of the objective as stated in the theorem, by use of Lemma 12.33. □

Remark 6.16. As in Remark 6.8, it is common to parameterize the inverse map $T_2^{-1}(y,\cdot;\theta)$ directly by letting $S_2(y,\cdot;\theta) \coloneqq T_2^{-1}(y,\cdot;\theta)$. This choice avoids inversion during learning, however it requires inverting the map $S_2$ to sample from the posterior $\pi^y$ after learning. ♢

The following result shows that the optimization problem in Theorem 6.15 over the space of inverse transports is convex for certain reference distributions.

**Theorem 6.17.** *If $\varrho_2$ is a log-concave reference density, then the optimization problem*

$$S_2^\star \in \arg\min_{S_2 \in \mathcal{S}} -\mathbb{E}^{(y,u)\sim\gamma}\Big[\log \varrho_2 \circ S_2(y,u) + \log\det D_u S_2(y,u)\Big], \qquad (6.14)$$

*over the space of invertible and diffeomorphic maps*

$$\mathcal{S} = \Big\{S_2 \in C^1(\mathbb{R}^{k+d};\mathbb{R}^d), D_u S_2(y,u) \succ 0 \text{ for all } u \in \mathbb{R}^d, y \in \mathbb{R}^k\Big\}$$

*is convex.*

*Proof.* The proof follows the steps of Theorem 5.3 by replacing the posterior $\pi^y$ with the reference density $\varrho_2$. □

Remark 6.18. Theorem 6.17 does not imply that the optimization problem in Theorem 6.15 is convex in $\theta$. In particular when $S$ has a non-linear parameterization based on neural networks, the problem will be non-convex. ♢

**Example 6.19.** Let $\varrho_2(u) = (2\pi)^{-d/2}\exp(-\frac{1}{2}|u|^2)$ be the standard Gaussian reference density. Then, $\log\varrho_2(u) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}|u|^2$. Ignoring the first term that is a constant, Theorem 6.17 shows that optimal map $S_2$ is found as solution of the optimization problem

$$\arg\min_{S_2} \mathbb{E}^{(y,u)\sim\gamma}\left[\frac{1}{2}|S_2(y,u)|^2 - \log\det D_u S_2(y,u)\right].$$

By Theorem 6.17, the optimization problem for $S_2$ is convex. The first term in the objective minimizes the squared norm of $S_2(y,u)$, which encourages the map's output to be at zero, the MAP point of the reference distribution $\varrho_2$. The second term prevents the map from concentrating the output at a single point. Moreover, the second term acts a log-barrier for the space of invertible maps by adding a large penalty as the derivative of $S_2$ approaches zero. ♢

The convexity of the objective yields uniqueness of the solution to (6.14) when a minimizer exists. The following result provides a concrete example for the closed-form solution when $\mathcal{S}$ is restricted to the space of affine maps.

**Theorem 6.20.** *Let $\varrho_2(u) = (2\pi)^{-d/2}\exp(-\frac{1}{2}|u|^2)$ be Gaussian and let $(y,u) \sim \gamma$. We consider $\mathcal{S}$ be the space of affine maps*

$$\mathcal{S} = \Big\{S_2(y,u) = A(u + By + c), A \in \mathbb{R}^{d\times d}, A \succ 0, B \in \mathbb{R}^{d\times k}, c \in \mathbb{R}^d\Big\},$$

*where $A$ is also constrained to be a triangular matrix. Then, the optimal map in the sense of solving (6.14) has the form*

$$S_2(y,u) = \Sigma_{u|y}^{-1/2}\Big(u - \mathbb{E}[u] + \Sigma_{uy}\Sigma_{yy}^{-1}(y - \mathbb{E}[y])\Big),$$

*where $\Sigma_{u|y} := \Sigma_{uu} - \Sigma_{uy}\Sigma_{yy}^{-1}\Sigma_{yu}$. The matrices $\Sigma_{uu}$, $\Sigma_{yy}$ and $\Sigma_{uy}$ denote the covariance of $u$, covariance of $y$, and cross-covariance of $(u,y)$ under $\gamma$, respectively.*

*Proof.* The optimization problem has the form

$$(A^\star, B^\star, c^\star) \in \arg\min_{A,B,c} \mathbb{E}^{(y,u)\sim\gamma}\left[\frac{1}{2}(u + By + c)^\top A^\top A(u + By + c) - \log\det A\right].$$

For fixed $A$ and $B$, taking the gradient of the objective with respect to $c$ and setting it equal to zero, we have

$$c^\star = -\mathbb{E}[u] - B\mathbb{E}[y].$$

Substituting the optimal $c^\star$ in the objective, we then define the loss for $B$ given a fixed $A$ to be

$$\mathsf{L}(B; A) := \mathbb{E}^{(y,u)\sim\gamma}\left[\frac{1}{2}\big(u - \mathbb{E}[u] + B(y - \mathbb{E}[y])\big)^\top A^\top A\big(u\mathbb{E}[u] + B(y - \mathbb{E}[y]) - \log\det A\right].$$

Taking the gradient with respect to $B$ and setting it equal to zero we have $D_B\mathsf{L}(B; A) = 2A^\top A(B\Sigma_{yy} + \Sigma_{uy}) = 0$, where $\Sigma_{yy}$ and $\Sigma_{uy}$ denote the covariance and cross-covariance of $y$ and $(u, y)$. Re-arranging for $B$ gives us

$$B^\star = -\Sigma_{uy}\Sigma_{yy}^{-1}.$$

Substituting the optimal $B^\star$ in the loss $\mathsf{L}$, we notice that the optimization problem for $A^\top A$ corresponds to the Gaussian variational inference problem in (2.7). The optimal solution for $A^\star$ is given by the inverse Cholesky factor of the covariance matrix

$$(A^\star)^{-1}(A^\star)^{-\top} = \Big(\mathbb{E}\Big[\big(u - \mathbb{E}[u] - \Sigma_{uy}\Sigma_{yy}^{-1}(y - \mathbb{E}[y])\big)\big(u - \mathbb{E}[u] - \Sigma_{uy}\Sigma_{yy}^{-1}(y - \mathbb{E}[y])\big)^\top\Big]\Big)^{-1}.$$

Expanding the squared terms yields

$$\begin{aligned}(A^\star)^{-1}(A^\star)^{-\top} &= \Sigma_{uu} - \Sigma_{uy}\Sigma_{yy}^{-1}\Sigma_{yu} - \Sigma_{uy}\Sigma_{yy}^{-1}\Sigma_{yu} + \Sigma_{uy}\Sigma_{yy}^{-1}\Sigma_{yy}\Sigma_{yy}^{-1}\Sigma_{yu}\\ &= \Sigma_{uu} - \Sigma_{uy}\Sigma_{yy}^{-1}\Sigma_{yu}\\ &= \Sigma_{u|y},\end{aligned}$$

which gives us the final result, $(A^\star)^{-1} = \Sigma_{u|y}^{1/2}$. □

## 6.3 Learning Likelihood Models

The framework in Section 6.2 can also be used to learn the likelihood function in an inverse problem on the basis of data. We again work in the setting of Assumption 6.4. Learning the likelihood is particularly useful in inverse problems where the likelihood is unknown analytically or intractable to evaluate as in Example 6.6. Given an approximation to the likelihood function and a known prior density, we have access to the approximate posterior density up to a normalizing constant. The resulting density can be used in algorithms such as Markov chain Monte Carlo to sample from the posterior distribution.

For this purpose we define a block-triangular map $T$, identical to the form in (6.5), but with a reversed ordering for the variables $(u, y)$; that is

$$T(u,y) = \begin{bmatrix} T_1(u) \\ T_2(u,y) \end{bmatrix}. \tag{6.15}$$

We use a map of this form, assumed for now to exactly pushforward the product reference density $\varrho(y,u) := \varrho_1(u)\varrho_2(y)$ to $\gamma(y,u)$; later we approximate this pushforward. In this section we factor $\gamma(y,u) = \rho(u)\mathsf{l}(y|u)$. By Theorem 6.9, with the roles of $u$ and $y$ reversed, the map $y \mapsto T_2(T_1^{-1}(u), y)$ pushes forward $\varrho_2(y)$ to $\mathsf{l}(y|u)$ for each $u$. We choose the first marginal of the reference density to be the prior, $\varrho_1(u) = \rho(u)$. Then, letting $T_1(u) = u$, the likelihood function is given by

$$\mathsf{l}(y|u) = \varrho_2(T_2^{-1}(u,y)) \det D_y T_2^{-1}(u,y), \tag{6.16}$$

where $T_2^{-1}(u,\cdot)$ denotes the inverse of the function $y \mapsto T_2(u,y)$.

**Example 6.21.** Let $\varrho_2(y)$ be the standard Gaussian on $\mathbb{R}^k$. Choosing $T_2(u,y) = G(u) - \Gamma^{1/2}y$ for some map $G\colon \mathbb{R}^d \to \mathbb{R}^d$ and a positive definite matrix $\Gamma \in \mathbb{R}^k \times \mathbb{R}^k$, we have the Gaussian likelihood function

$$\begin{aligned} \mathsf{l}(y|u) &= \varrho_2\Big(\Gamma^{-1/2}(y - G(u))\Big) \det \Gamma^{-1/2} \\ &= \frac{1}{\sqrt{(2\pi)^k \det \Gamma}} \exp\Big(-\frac{1}{2}|y - G(u)|_\Gamma^2\Big) \\ &= \mathcal{N}(G(u), \Gamma). \end{aligned}$$

◇

Following the approach in Subsection 6.2.2, we learn the parameters $\theta$ to define a map $T(u,y;\theta)$ which approximates the exact pushforward. Thus we solve the optimization problem

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{D}_{\mathrm{KL}}(\gamma \| T(\cdot;\theta)_\sharp \varrho). \tag{6.17}$$

The next theorem shows that we can find the parameters $\theta^\star$ by minimizing a loss function that only depends on the joint distribution $\gamma$ via an expectation, and thus is amenable to learning from paired data. The proof is analogous to that of Theorem 6.15 but with the roles of $y$ and $u$ reversed.

**Theorem 6.22.** *Let $T$ be a block-triangular transport map of the form in* (6.15)*. Let the reference density be $\varrho(u,y) = \rho(u)\varrho_2(y)$ and $T_1(u) = u$. The optimal parameters $\theta^\star$ from* (6.17) *also solve the optimization problem*

$$\theta^\star \in \arg\min_{\theta\in\Theta} -\mathbb{E}^{(y,u)\sim\gamma}\Big[\log \varrho_2 \circ T_2^{-1}(u,y;\theta) + \log\det D_y T_2^{-1}(u,y;\theta)\Big]. \tag{6.18}$$

*Proof.* From the chain rule for the KL divergence, Lemma 12.29, we have

$$\mathsf{D}_{\mathrm{KL}}(\gamma \| T(\cdot;\theta)_\sharp \varrho) = \mathsf{D}_{\mathrm{KL}}(\rho \| (T_1)_\sharp \rho) + \mathbb{E}^{u\sim\rho}\big[\mathsf{D}_{\mathrm{KL}}(\mathsf{l}(\cdot|u) \| T_2(u,\cdot;\theta)_\sharp \varrho_2)\big]. \tag{6.19}$$

Setting $T_1(u) = u$, the first term in (6.19) is zero, and the second term is given by

$$\begin{aligned}\mathbb{E}^{u\sim\rho}[\mathsf{D}_{\mathrm{KL}}(\mathsf{l}(\cdot|u) \| T_2(u,\cdot;\theta)_\sharp \varrho_2)] &= \int \rho(u)\mathsf{l}(y|u)\Big[\log \mathsf{l}(y|u) - \log\big(T_2(u,\cdot;\theta)_\sharp \varrho_2(y)\big)\Big]\, dydu \\ &= c - \int \gamma(y,u)\log\big(T_2(u,\cdot;\theta)_\sharp \varrho_2(y)\big)\, dudy,\end{aligned}$$

where $c$ is a constant that is independent of $\theta$. Therefore, minimizing the second term achieves the minimum of the objective as stated in the theorem, by use of Lemma 12.33. □

Given a collection of $N$ paired samples $\{(y^{(n)}, u^{(n)})\}_{n=1}^N$ from the joint distribution $\gamma(y,u)$ we may approximate the expectation in (6.18). This leads to the problem of minimizing the following empirical loss function to find the approximate likelihood

$$\arg\min_{\theta\in\Theta} -\frac{1}{N}\sum_{n=1}^N \log \varrho_2 \circ T_2^{-1}(u^{(n)}, y^{(n)};\theta) + \log\det D_y T_2^{-1}(u^{(n)}, y^{(n)};\theta). \tag{6.20}$$

**Example 6.23.** Let $\varrho_2$ be a standard Gaussian density of dimension $\mathbb{R}^k$ and let $T_2$ be a parameterized transport map of the form $T_2(u,y;\theta) = G(u;\theta) - \Gamma^{1/2}y$ as in Example 6.21 where $\Gamma \succ 0$ is known. Then, minimizing the loss function in (6.20) corresponds to solving the problem

$$\begin{aligned}&\arg\min_{\theta\in\Theta}\left\{\frac{1}{N}\sum_{n=1}^N \frac{1}{2}\Big|T_2^{-1}(u^{(n)}, y^{(n)};\theta)\Big|^2 + \log\det D_y T_2^{-1}(u^{(n)}, y^{(n)};\theta)\right\} \\ &= \arg\min_{\theta\in\Theta}\left\{\frac{1}{N}\sum_{n=1}^N \frac{1}{2}\Big|\Gamma^{-1/2}(y^{(n)} - G(u^{(n)};\theta))\Big|_\Gamma^2 + \log\det \Gamma^{-1/2}\right\} \\ &= \arg\min_{\theta\in\Theta} \frac{1}{N}\sum_{n=1}^N \frac{1}{2}\Big|y^{(n)} - G(u^{(n)};\theta)\Big|_\Gamma^2.\end{aligned}$$

Hence, finding the transport map is equivalent to seeking an approximate forward model as the solution of a mean-squared regression problem. This generalizes the approach of learning forward surrogate models in Chapter 3. ♢

## 6.4 Amortized MAP Estimation

Theorem 1.2 delivers the posterior

$$\pi^y(u) = \frac{1}{Z}\nu(y - G(u))\rho(u).$$

Recall that the MAP estimator of $u$ given data $y$ is defined as any point solving the maximization problem

$$u_{\mathrm{MAP}} \in \arg\max_{u\in\mathbb{R}^d} \pi^y(u).$$

We make the following assumption:

**Data Assumption 6.24.** *We are given multiple pairs of observations and numerically computed MAP estimators* $\{(y^{(n)}, u^{(n)}_{\mathrm{MAP}})\}_{n=1}^N$, *drawn i.i.d. from probability measure* $\mu$ *on* $\mathbb{R}^k \times \mathbb{R}^d$.

Remark 6.25. The measure $\mu$ may be constructed as the product of a measure on $\mathbb{R}^k$ and its pushforward under a deterministic algorithm to find the MAP estimator in $\mathbb{R}^d$ from data in $\mathbb{R}^k$; however, as the MAP estimation algorithm may be random (for example stochastic gradient descent) we work with a general measure $\mu$. ♢

Given measure $\mu$ we may try and learn the dependence of the MAP estimator on observations. To this end we seek a parameterized function $u\colon \mathbb{R}^k \times \Theta \to \mathbb{R}^d$ for $\Theta \subseteq \mathbb{R}^p$. We aim to find $\theta^\star \in \Theta$ for which $u(\cdot;\theta^\star) : \mathbb{R}^k \to \mathbb{R}^d$ approximates the mapping from observed data to the MAP estimator. Recall Definition 12.56 from Subsection 12.3.7 on distance-like deterministic scoring rules $\mathsf{D}$. We may use this to compare the point estimator $u(y;\theta)$ at point $y$ to $u_{\mathrm{MAP}}$ obtained from that same $y$. We then define $\theta^\star$ by

$$\begin{aligned} \mathsf{J}(\theta) &= \mathbb{E}^{(y,u_{\mathrm{MAP}})\sim\mu}\big[\mathsf{D}\big(u(y;\theta), u_{\mathrm{MAP}}\big)\big],\\ \theta^\star &\in \arg\min_{\theta\in\Theta} \mathsf{J}(\theta). \end{aligned} \tag{6.21}$$

In practice we replace the expectation over $\mu$ in (6.21) with expectation with respect to $\mu^N$, the empirical measure defined by $\{(y^{(n)}, u^{(n)}_{\mathrm{MAP}})\}_{n=1}^N$.

## 6.5 Bibliography

The variable $z$ over which we marginalize is often referred to as a *nuisance* or auxiliary random variable; this is often used in the setting where $z$ is not the primary parameter of interest in the inverse problem. An alternative approach to those described here seeks the joint posterior of $(u, z)$, and then marginalizes after the computation. While the corresponding likelihood for the joint posterior doesn't involve a marginal or integrated likelihood, it can lead to a challenging problem for high or even infinite-dimensional latent variables. We refer to [24] for more details on integrated likelihoods. In MCMC algorithms, these integrated likelihoods are commonly addressed using pseudo-marginal methods that work with unbiased estimators of the likelihood [11].

Approximate Bayesian computation (ABC) is a classic inference method for performing likelihood-free inference with latent variables or other intractable likelihoods; see [289] for a comprehensive overview on ABC. These approaches define a distance function that compare simulated observations to the true observation and reject parameter samples that are not consistent with the true observation based on a small tolerance. ABC methods can be shown to be consistent in the limit of the tolerance approaching zero, but typically require large sample sizes with high-dimensional observations.

Learning approaches have appeared as alternatives to ABC; machine learning-based approaches for simulation-based inference are outlined in [65]. Methodologies for both posterior and likelihood approximation in this setting are implemented using various unsupervised learning architectures (see Chapter 13) including: conditional normalizing flows [335, 243], conditional generative adversarial networks [267], conditional diffusion models [25]. Triangular transport maps are a core element of these architectures for solving inverse problems. They are related to the well known Knothe-Rosenblatt transport, which has been used for conditional density estimation in [212, 23]. The design and analysis of block-triangular maps on function space is investigated in [21, 151].

Amortized inference is overviewed in [339]. Variational autoencoders, which were proposed concurrently in [173] and [271], introduced the concept of amortized inference in the specific context of autoencoding. The idea of iterating amortized inference was introduced in [210]; this approach is designed to close the amortization gap caused by failing to reach optimality when training.

# Part II

# Data Assimilation

# Chapter 7

## Filtering and Smoothing Problems

This chapter is devoted to *data assimilation* problems. We study both the *filtering problem* and the *smoothing problem.* Consider the *stochastic dynamics model* given by

$$v^\dagger_{j+1} = \Psi(v^\dagger_j) + \xi^\dagger_j, \quad j \in \mathbb{Z}^+, \tag{7.1a}$$

$$v^\dagger_0 \sim \mathcal{N}(m_0, C_0), \quad \xi^\dagger_j \sim \mathcal{N}(0, \Sigma) \text{ i.i.d.}, \tag{7.1b}$$

where we assume that the i.i.d. sequence $\{\xi^\dagger_j\}_{j\in\mathbb{Z}^+}$ is independent of initial condition $v^\dagger_0$; this is often written as $\{\xi^\dagger_j\}_{j\in\mathbb{Z}^+} \perp\!\!\!\perp v^\dagger_0$. The *data model* is given by

$$y^\dagger_{j+1} = h(v^\dagger_{j+1}) + \eta^\dagger_{j+1}, \quad j \in \mathbb{Z}^+, \tag{7.2a}$$

$$\eta^\dagger_j \sim \mathcal{N}(0, \Gamma) \quad \text{i.i.d.}, \tag{7.2b}$$

where we assume that the i.i.d. sequence $\{\eta^\dagger_j\}_{j\in\mathbb{N}} \perp\!\!\!\perp v^\dagger_0$ and that the two i.i.d. sequences in the dynamics and data models are independent of one another: $\{\xi^\dagger_j\}_{j\in\mathbb{Z}^+} \perp\!\!\!\perp \{\eta^\dagger_j\}_{j\in\mathbb{N}}$.

Broadly speaking, data assimilation seeks to find the state $\{v^\dagger_j\}$, over some set of time indices $j \in \{0, 1, \dots, J\}$ based on realized observations $\{y^\dagger_j\}$ from (7.2). We define, for a given and fixed integer $J$,

$$V^\dagger := \{v^\dagger_0, \dots, v^\dagger_J\}, \; Y^\dagger := \{y^\dagger_1, \dots, y^\dagger_J\}, \text{ and } Y^\dagger_j := \{y^\dagger_1, \dots, y^\dagger_j\}. \tag{7.3}$$

There are two core problems in data assimilation: one known as *filtering*, and the other as *smoothing*. The goal of the filtering problem is to approximate the sequence of probability distributions, $\mathbb{P}(v^\dagger_j | Y^\dagger_j)$, for the state of the system $v^\dagger_j$ at index $j$ conditioned on data $Y^\dagger_j := \{y^\dagger_1, \dots, y^\dagger_j\}$; and furthermore to do so as it arrives sequentially. The smoothing problem, in contrast, is to estimate the probability distribution on an entire sequence of states $V^\dagger$ given the entire dataset $Y^\dagger$, $\mathbb{P}(V^\dagger | Y^\dagger)$. The sequence $V^\dagger$ is often termed the *signal* and the sequence $Y^\dagger$ the *data*. Note that $V^\dagger \in \mathbb{R}^{d(J+1)}$ and $Y^\dagger \in \mathbb{R}^{kJ}$ represent sequences over a window of length $J+1$ and $J$, respectively.

Remark 7.1. When trying to solve these probabilistic filtering and smoothing problems, we will refer to *probabilistic estimation.* On the other hand many optimization-based filtering and smoothing algorithms exist with the aim being simply to estimate the state itself, conditioned on data. We refer to this as *state estimation.* ♢

Remark 7.2. Because filtering is defined by conditioning on data arriving sequentially, such algorithms may be used *online*, the probability distribution over states being updated every time a new data point arrives. Smoothing gives rise to methodologies that are most naturally used, in their most basic form, in an *offline* fashion. However, smoothing algorithms may also be used sequentially, in block form with respect to discrete time index $j$, as we now explain.

In the above we assumed data $y_j^\dagger$ given on the index set $j \in \{1, \dots, J\}$. Now let us assume that the data $y_j^\dagger$ is in fact given for all $j \in \mathbb{N}$ and use integer $J$ to break up the data into finite time segments. We first solve the smoothing problem defined on the index set $j \in \{0, \dots, J\}$. Taking the solution at time $J$ as starting point we may then solve a smoothing problem on index set $j \in \{J, \dots, 2J\}$. This idea may be iterated, working on index set $j \in \{kJ, \dots, (k+1)J\}$, then on index set $j \in \{(k+1)J, \dots, (k+2)J\}$ and so on. Such methods are known as *fixed-interval smoothers*. At overlap points, which are a multiple of $J$, some form of Gaussian projection will be needed to restart the process on the next time-interval as the assumption made above is that the initial condition is Gaussian; however, this can be relaxed. $\diamondsuit$

Before describing the filtering and smoothing problems we outline assumptions that are made in the remainder of the chapters devoted to data assimilation.

**Assumption 7.3.** *The matrices $C_0$, $\Sigma$ and $\Gamma$ are positive definite. Furthermore the nonlinear maps $\Psi$ and $h$ are continuous: $\Psi \in C(\mathbb{R}^d, \mathbb{R}^d)$ and $h \in C(\mathbb{R}^d, \mathbb{R}^k)$.*

In Sections 7.1 and 7.2, respectively, we formulate the filtering and smoothing problems. Section 7.3 recalls the Kalman filter, applicable in the linear and Gaussian setting. Sections 7.4, 7.5, 7.6, and 7.7 recall 3DVar, the extended Kalman filter (ExKF), and the ensemble Kalman filter (EnKF); Sections 7.7 and 7.8 concern the bootstrap particle filter and the optimal particle filter approaches to filtering. Section 7.9 is devoted to the 4DVar approach to smoothing, followed by a discussion of reanalysis in Section 7.10. In Section 7.11 we discuss model error in the context of filtering and smoothing problems. Section 7.12 contains bibliographic notes.

Remark 7.4. We conclude this introductory discussion with some remarks on generalizations of the setting we adopt in these notes. First we observe that the stochastic dynamics model (7.1) and data model (7.2) are readily generalized to settings in which the maps $\Psi(\cdot)$ and $h(\cdot)$, as well as the covariances $\Sigma$ and $\Gamma$, are dependent on the time-index $j$. It is also possible to allow degenerate noises (positive semi-definite), and the case where no noise is present in the dynamics ($\Sigma \equiv 0$) arises frequently. Furthermore, all of the sections in this chapter, with the exception of the Kalman filter which leverages Gaussian structure, admit generalizations to settings in which the noises $\{\xi_j^\dagger\}_{j\in\mathbb{Z}^+}, \{\eta_j^\dagger\}_{j\in\mathbb{N}}$ are independent i.i.d. centred but *non-Gaussian* sequences. Adding *correlation* to the noises is also possible. For example it is possible to consider correlation across the discrete time index $j$. This significantly complicates filtering, but it is possible to accommodate it if the noise itself is generated by a Markov process. Adding correlation to the noises across the discrete time index $j$ also complicates smoothing, but can also be handled, typically at additional computational cost. In addition, allowing for correlation between the dynamics and observational noise is also possible. Finally we note that continuity of the maps $\Psi(\cdot), h(\cdot)$ is also not necessary. $\diamondsuit$

## 7.1 Formulation of the Filtering Problem

**Definition 7.5.** The *filtering problem* is to find, and update sequentially in $j$, the probability densities $\pi_j(v_j^\dagger) := \mathbb{P}(v_j^\dagger|Y_j^\dagger)$ on $\mathbb{R}^d$ for $j = 1, \ldots, J$. We refer to $\pi_j$ as the *filtering distribution at time $j$.* $\diamondsuit$

Filtering may be understood as the sequential interleaving of prediction, using the stochastic dynamics model (7.1), with inversion, using the data model (7.2). To explain this perspective it is helpful to introduce $\widehat{\pi}_{j+1} = \mathbb{P}(v_{j+1}^\dagger|Y_j^\dagger)$ and to decompose the sequential updating $\pi_j \mapsto \pi_{j+1}$ into the following two steps:

$$\begin{aligned} &\textbf{Prediction Step:} \qquad \widehat{\pi}_{j+1} = \mathsf{P}\pi_j.\\ &\textbf{Analysis Step:} \qquad \pi_{j+1} = \mathsf{A}(\widehat{\pi}_{j+1}; y_{j+1}^\dagger). \end{aligned} \tag{7.4}$$

The combination of the prediction and analysis steps is shown schematically in Figure 7.1 and leads to the update

$$\pi_{j+1} = \mathsf{A}(\mathsf{P}\pi_j; y_{j+1}^\dagger). \tag{7.5}$$

Here $\mathsf{P}$ is a linear map, defining the Markov process underlying the stochastic dynamics model and $\mathsf{A}(\cdot, y_{j+1}^\dagger)$ is a nonlinear likelihood map defined by application of Bayes Theorem 1.2 to solve the inverse problem for $v_{j+1}^\dagger|y_{j+1}^\dagger$, with prior $\widehat{\pi}_{j+1}$.

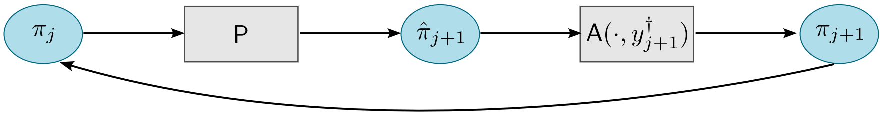

**Figure 7.1** Prediction and analysis steps combined.

The linear operator $\mathsf{P}$ is defined as follows:

$$\mathsf{P}\pi(u) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma}} \int \exp\Bigl(-\frac{1}{2}|u - \Psi(v)|_\Sigma^2\Bigr)\pi(v)\,dv. \tag{7.6}$$

Given any probability density function $\pi$ on $\mathbb{R}^d$ we can extend to a joint density $\nu_\pi$ on state-data space $\mathbb{R}^d \times \mathbb{R}^k$ :

$$\nu_\pi(u, y) = \frac{1}{\sqrt{(2\pi)^k \det \Gamma}} \exp\Bigl(-\frac{1}{2}|y - h(u)|_\Gamma^2\Bigr)\pi(u).$$

Using this we can define the nonlinear operator $\mathsf{A}(\cdot, y_{j+1}^\dagger)$ by

$$\mathsf{A}(\pi, y^\dagger)(u) = \frac{\nu_\pi(u, y^\dagger)}{\int_{\mathbb{R}^d} \nu_\pi(u, y^\dagger)\,du}. \tag{7.7}$$

The following theorem shows that the filtering distribution has an interesting property relating its mean to its variance; later we will use this to motivate study of the spread–error ratio of forecasts that have a probabilistic interpretation.

**Theorem 7.6.** *In the following the pair $(v_j^\dagger, Y_j^\dagger)$ is distributed according to the joint distribution under the dynamics/data model (7.1), (7.2), with expection denoted $\mathbb{E}^{(v_j^\dagger, Y_j^\dagger)}$. Furthermore $v_j^\dagger$ and $Y_j^\dagger$ are distributed according to their respective marginals with respect to this distribution, with expectation under the latter denoted $\mathbb{E}^{Y_j^\dagger}$. And $v_j$ is distributed according to the conditional $v_j^\dagger|Y_j^\dagger$, the filtering distribution with expectation denoted by $\mathbb{E}$. Then*

$$\frac{\mathbb{E}^{Y_j^\dagger}\,\mathbb{E}\Big|v_j - \mathbb{E}v_j\Big|^2}{\mathbb{E}^{(v_j^\dagger, Y_j^\dagger)}\Big|v_j^\dagger - \mathbb{E}v_j\Big|^2} = 1.$$

*Proof.* This is a consequence of the properties of conditional probability, using that $\mathbb{P}(v_j^\dagger, Y_j^\dagger)$ can be factored as $\mathbb{P}(v_j^\dagger|Y_j^\dagger)\mathbb{P}(Y_j^\dagger)$. □

Remark 7.7. This theorem motivates discussion of the *spread-error ratio* in Chapter 9. This terminology is used because the denominator is the mean square *error* of the mean under the filtering distribution, with respect to the true signal underlying the data, averaged over the joint distribution of the model; the numerator is the variance under the filtering distribution (the *spread*), averaged over the marginal on the data. The theorem explains the sense in which the variance of the filter provides an estimate of the error made by using the mean under the filter. Note also that the preceding statement and proof can be generalized to the smoothing distribution $\mathbb{P}(v_j^\dagger|Y^\dagger)$ if $Y_j^\dagger$ is replaced by $Y^\dagger$. ♢

## 7.2 Formulation of the Smoothing Problem

Recall definitions of $V^\dagger, Y^\dagger$ and $Y_j^\dagger$ from (7.3). We note that the stochastic dynamics model (7.1) defines a probability distribution on $V^\dagger$, with density $\rho(V^\dagger)$, defined through the Markovian structure as

$$\rho(V^\dagger) \propto \exp\Big(-\frac{1}{2}|v_0^\dagger - m_0|_{C_0}^2 - \frac{1}{2}\sum_{j=0}^{J-1}|v_{j+1}^\dagger - \Psi(v_j^\dagger)|_\Sigma^2\Big). \tag{7.8}$$

**Definition 7.8.** The *smoothing problem* is to find the probability density $\Pi^{Y^\dagger}(V^\dagger) := \mathbb{P}(V^\dagger|Y^\dagger) = \mathbb{P}(\{v_0^\dagger, \dots, v_J^\dagger\}|Y_j^\dagger)$ in $\mathcal{P}(\mathbb{R}^{d(J+1)})$ for some fixed integer $J$. We refer to $\Pi^{Y^\dagger}$ as the *smoothing distribution.* ♢

To make connection with the Bayesian inverse problems of Chapter 1 we define

$$\eta^\dagger := \{\eta_1^\dagger, \dots, \eta_J^\dagger\} \in \mathbb{R}^{kJ},$$

noting that $\eta^\dagger \sim \mathcal{N}(0, \mathsf{\Gamma})$, where $\mathsf{\Gamma}$ is block diagonal with $\Gamma$ in each diagonal block. If we then define $\mathsf{h} : \mathbb{R}^{d(J+1)} \to \mathbb{R}^{kJ}$ by

$$\mathsf{h}(V^\dagger) := \{h(v_1^\dagger), \dots, h(v_J^\dagger)\}$$

then the data model may be written as

$$Y^\dagger = \mathsf{h}(V^\dagger) + \eta^\dagger.$$

We are interested in finding $\mathbb{P}(V^\dagger | Y^\dagger) = \Pi^{Y^\dagger}(V^\dagger)$. We may apply Bayes Theorem 1.2, with prior $\rho$ noting that, under this prior, $V^\dagger \perp\!\!\!\perp \eta^\dagger$, the standard setting for Bayesian inversion from Chapter 1. We obtain

$$\Pi^{Y^\dagger}(V^\dagger) \propto \exp\Big(-\frac{1}{2}|Y^\dagger - \mathsf{h}(V^\dagger)|_{\mathsf{\Gamma}}^2\Big)\rho(V^\dagger), \tag{7.9}$$

also a probability density function in $\mathcal{P}(\mathbb{R}^{d(J+1)})$. Recall from Chapter 1 the likelihood which here has the form

$$\mathsf{l}(Y^\dagger | V) \propto \exp\Big(-\frac{1}{2}|Y^\dagger - \mathsf{h}(V)|_{\mathsf{\Gamma}}^2\Big), \tag{7.10}$$

enabling us to write (7.9) in the form

$$\Pi^{Y^\dagger}(V) \propto \mathsf{l}(Y^\dagger | V)\rho(V). \tag{7.11}$$

Note that $V$ is a dummy variable, explaining why we have dropped the † on it, but not on the fixed instance of the data $Y^\dagger$.

## 7.3 Kalman Filter

Suppose that at time $j$, $\Psi(\cdot) = A_j \cdot$ and $h(\cdot) = H_j \cdot$: the model and observation operator are linear, but can change in time. Then the solution to the filtering problem is Gaussian and $\pi_j = \mathcal{N}(v_j, C_j)$, $\widehat{\pi}_{j+1} = \mathcal{N}(\widehat{v}_{j+1}, \widehat{C}_{j+1})$. The update rules for the mean and covariance are given by the *Kalman filter*: the mean updates according to

$$\widehat{v}_{j+1} = A_j v_j, \tag{7.12a}$$

$$v_{j+1} = \widehat{v}_{j+1} + K_{j+1}(y_{j+1}^\dagger - H_{j+1}\widehat{v}_{j+1}); \tag{7.12b}$$

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

$$\widehat{C}_{j+1} = A_j C_j A_j^\top + \Sigma, \tag{7.13a}$$

$$K_{j+1} = \widehat{C}_{j+1} H_{j+1}^\top (H_{j+1}\widehat{C}_{j+1}H_{j+1}^\top + \Gamma)^{-1}, \tag{7.13b}$$

$$C_{j+1} = (I - K_{j+1}H_{j+1})\widehat{C}_{j+1}. \tag{7.13c}$$

Note that the update for the covariance evolves independently of the update for the mean. Furthermore, the covariance update is independent of the data. To motivate 3DVar, which we describe in the next section, we note that if $A_j = A$ and $H_j = H$

are constant in time, and the covariance is in steady state, then $C_{j+1} = C_j = C_\infty$ and $\widehat{C}_{j+1} = \widehat{C}_j = \widehat{C}_\infty$. Under appropriate controllability and observability assumptions, the steady-state covariance and gain can be obtained by finding the unique solution $(\widehat{C}_\infty, K_\infty)$ to the equations

$$\widehat{C}_\infty = A(I - K_\infty H)\widehat{C}_\infty A^\top + \Sigma, \tag{7.14a}$$

$$K_\infty = \widehat{C}_\infty H^\top (H\widehat{C}_\infty H^\top + \Gamma)^{-1}, \tag{7.14b}$$

and setting

$$C_\infty = (I - K_\infty H)\widehat{C}_\infty. \tag{7.15}$$

Using the steady-state Kalman gain from (7.14), the state updates become

$$\widehat{v}_{j+1} = Av_j, \tag{7.16a}$$

$$v_{j+1} = \widehat{v}_{j+1} + K_\infty (y^\dagger_{j+1} - H\widehat{v}_{j+1}). \tag{7.16b}$$

We note the form of this update of the mean $v_j \mapsto v_{j+1}$ : it comprises a prediction step $v_j \mapsto \widehat{v}_{j+1}$, derived from the stochastic dynamics model, and an analysis step $\widehat{v}_{j+1} \mapsto v_{j+1}$ defined by the data model.

## 7.4 3DVar

Motivated by the form (7.16) of the mean update equations for the Kalman filter, when the gain is in steady state, we propose the following generalization to the setting of nonlinear $(\Psi, h)$ :

$$\widehat{v}_{j+1} = \Psi(v_j) + s\xi_j, \tag{7.17a}$$

$$v_{j+1} = \widehat{v}_{j+1} + K(y^\dagger_{j+1} - h(\widehat{v}_{j+1})). \tag{7.17b}$$

The most basic form will use $s = 0$; this is then consistent with the Kalman filter with frozen gain (7.16). If $s = 1$ is used, to mimic the true forward model, then $\xi_j \sim \mathcal{N}(0, \Sigma)$ is an i.i.d. sequence.

Remark 7.9. We make some comments on the terminology 3DVar that we employ here. The approach is in fact more properly termed *cycled 3DVar*. In this context the *3DVar* component of the nomenclature refers to the analysis step (7.17b), and its formulation via an optimization problem, which we detail below; the *cycled* component of the nomenclature refers to repeatedly solving this optimization problem, as each data point is acquired, and interleaving this with the prediction step (7.17a). The use of 3D refers to the fact that, in weather forecasting, the optimization problem is for a field in three physical space dimensions.

To appreciate this connection to optimization we consider the case where $\Psi$ is allowed to be nonlinear, but $h(\cdot) = H\cdot$ is linear. The predict-then-optimize viewpoint then leads to 3DVar above if $v_{j+1}$ is computed from $(\widehat{v}_{j+1}, y^\dagger_{j+1})$ by solving the optimization problem

$$\mathsf{J}_j(v) = \frac{1}{2}|v - \widehat{v}_{j+1}|^2_{\widehat{C}} + \frac{1}{2}|y^\dagger_{j+1} - Hv|^2_\Gamma, \tag{7.18a}$$

$$v_{j+1} \in \arg\min_{v \in \mathbb{R}^d} \mathsf{J}_j(v). \tag{7.18b}$$

This minimization is equivalent to (7.17b), in the case $h(\cdot) = H\cdot$, provided that

$$K = \widehat{C}H^\top (H\widehat{C}H^\top + \Gamma)^{-1}. \tag{7.19}$$

In particular this indicates a methodology for choosing $K$: instead choose an estimate of the uncertainty in the prediction, $\widehat{C}$, and use this to define $K$. This approach to modeling $\widehat{C}$, and then deducing $K$, is natural because the uncertainty in the prediction is an interpretable quantity. ♢

## 7.5 Extended Kalman Filter (ExKF)

The extended Kalman filter (ExKF) generalizes the Kalman filter to nonlinear dynamics and observation operators by linearizing these functions in order to compute the gain and propagate the covariance. We define the Jacobians of these functions and evaluate them at the outputs $v_j$ and $\widehat{v}_j$ of a putative filtering algorithm:

$$A_j := D\Psi(v_j), \tag{7.20a}$$
$$H_j := Dh(\widehat{v}_j). \tag{7.20b}$$

These Jacobians may be known analytically; otherwise, auto-differentiation (Section 16.2) can be used to obtain them.

The mean update is then done using the nonlinear $\Psi$ and $h$, similarly to 3DVar (7.17) with $s = 0$ and with an evolving gain:

$$\widehat{v}_{j+1} = \Psi(v_j), \tag{7.21a}$$
$$v_{j+1} = \widehat{v}_{j+1} + K_j\big(y^\dagger_{j+1} - h(\widehat{v}_{j+1})\big). \tag{7.21b}$$

The computation of $K_j$, and the covariances involved in its definition, uses (7.13a), but with evolving $(A_j, H_j)$ defined by (7.20):

$$\widehat{C}_{j+1} = A_j C_j A_j^\top + \Sigma, \tag{7.22a}$$
$$K_{j+1} = \widehat{C}_{j+1}H_{j+1}^\top\big(H_{j+1}\widehat{C}_{j+1}H_{j+1}^\top + \Gamma\big)^{-1}, \tag{7.22b}$$
$$C_{j+1} = (I - K_{j+1}H_{j+1})\widehat{C}_{j+1}. \tag{7.22c}$$

Remark 7.10. The extended Kalman filter recovers the true filtering distribution in the case of linear $\Psi$ and $h$, in which case it reduces to the standard Kalman filter. It can often perform well beyond the linear Gaussian setting: when $\Psi$ and $h$ are close to linear; or when small covariances are assumed in both the dynamics and data models, in order to justify linearization of the predictive and analysis distributions about the mean. ♢

Despite the simplicity of the ExKF, computing covariances when $d$ or $k$ is large can be prohibitively expensive. This motivates the ensemble methods we describe next.

## 7.6 Ensemble Kalman Filter (EnKF)

### 7.6.1 The EnKF Algorithm

The content of Remark 7.9, concerning 3DVar, is to shift the problem of choosing $K$ to one of choosing $\widehat{C}$, an estimate of the uncertainty in the predictions. In Remark 7.10, concerning ExKF, it is discussed how this may be achieved via linearization. The Ensemble Kalman Filter (EnKF) builds on this idea by running an ensemble of $N$ 3DVar-like algorithms, with a time dependent gain $K_{j+1}$, estimated via empirical estimates of the covariances of the predicted states $\{\widehat{v}_{j+1}^{(n)}\}_{n=1}^N$ and $\{h(\widehat{v}_{j+1})^{(n)}\}_{n=1}^N$, their mappings under $h$.

The ensemble $\{v_j^{(n)}\}_{n=1}^N \mapsto \{v_{j+1}^{(n)}\}_{n=1}^N$ according to the following algorithm:

$$\widehat{v}_{j+1}^{(n)} = \Psi(v_j^{(n)}) + \xi_j^{(n)}, \quad n = 1, \dots, N, \tag{7.23a}$$

$$v_{j+1}^{(n)} = \widehat{v}_{j+1}^{(n)} + K_{j+1}\big(y_{j+1}^\dagger - \eta_{j+1}^{(n)} - h(\widehat{v}_{j+1}^{(n)})\big), \quad n = 1, \dots, N. \tag{7.23b}$$

Here $\xi_j^{(n)} \sim \mathcal{N}(0, \Sigma), \quad \eta_{j+1}^{(n)} \sim \mathcal{N}(0, \Gamma)$ are independent sequences of i.i.d. random vectors with respect to both $j$ and $n$, and the two sequences themselves are independent of one another. The gain matrix $K_{j+1}$ is calculated according to

$$\widehat{m}_{j+1} = \frac{1}{N}\sum_{n=1}^N \widehat{v}_{j+1}^{(n)}, \tag{7.24a}$$

$$\widehat{h}_{j+1} = \frac{1}{N}\sum_{n=1}^N h(\widehat{v}_{j+1}^{(n)}), \tag{7.24b}$$

$$\widehat{C}_{j+1}^{vh} = \frac{1}{N}\sum_{n=1}^N (\widehat{v}_{j+1}^{(n)} - \widehat{m}_{j+1}) \otimes (h(\widehat{v}_{j+1}^{(n)}) - \widehat{h}_{j+1}), \tag{7.24c}$$

$$\widehat{C}_{j+1}^{hh} = \frac{1}{N}\sum_{n=1}^N (h(\widehat{v}_{j+1}^{(n)}) - \widehat{h}_{j+1}) \otimes (h(\widehat{v}_{j+1}^{(n)}) - \widehat{h}_{j+1}), \tag{7.24d}$$

$$\widehat{C}_{j+1}^{yy} = \widehat{C}_{j+1}^{hh} + \Gamma, \quad K_{j+1} = \widehat{C}_{j+1}^{vh}(\widehat{C}_{j+1}^{yy})^{-1}. \tag{7.24e}$$

Remark 7.11. We note that in the case where $h(\cdot) = H\cdot$ (and is hence linear) we may calculate the gain through estimation of a single covariance, as follows:

$$\widehat{C}_{j+1} = \frac{1}{N}\sum_{n=1}^N (\widehat{v}_{j+1}^{(n)} - \widehat{m}_{j+1}) \otimes (\widehat{v}_{j+1}^{(n)} - \widehat{m}_{j+1}), \tag{7.25a}$$

$$K_{j+1} = \widehat{C}_{j+1}H^\top(H\widehat{C}_{j+1}H^\top + \Gamma)^{-1}. \tag{7.25b}$$

This should be compared with (7.19), used in 3DVar. The analysis mean and covariance can be estimated as

$$m_{j+1} = \frac{1}{N}\sum_{n=1}^{N} v_{j+1}^{(n)}, \tag{7.26a}$$

$$C_{j+1} = \frac{1}{N}\sum_{n=1}^{N} (v_{j+1}^{(n)} - m_{j+1}) \otimes (v_{j+1}^{(n)} - m_{j+1}). \tag{7.26b}$$

♢

The ensemble may be used to furnish an approximation of the filtering distribution by simply forming an empirical measure from the ensemble members: we approximate $\pi_j \approx \pi_j^{\mathrm{EnKF}}$ where

$$\pi_j^{\mathrm{EnKF}} = \frac{1}{N}\sum_{n=1}^{N} \delta_{v_j^{(n)}}. \tag{7.27}$$

Remark 7.12. Conditions under which (7.27) provides a good approximation of the true filtering distribution are discussed in the bibliography Section 7.12, and revolve around Gaussian approximations. When the true filtering distribution is far from Gaussian the method will not provide a good approximation.

However ensemble Kalman filters are widely used in high dimensional problems because they do not suffer from the weight collapse that plagues the particle filters of Sections 7.7 and 7.8. Ensemble Kalman filters often need inflation and localization to perform well for high-dimensional systems, for chaotic dynamical systems, and for small ensemble sizes. Inflation and localization are discussed in the following two subsections. ♢

### 7.6.2 Inflation

For simplicity we confine the discussion to the case of linear observation operator $h(\cdot) = H\,\cdot$ . The ensemble Kalman gain is then determined by $\widehat{C}_{j+1}$, as explained in Remark 7.11. We assume that the covariance at infinite sample size $N = \infty$ leads to the optimal gain; this is provably true for linear $\Psi$, as discussed in the bibliography. From this perspective, ensemble Kalman filters suffer from sampling error due to finite ensemble size $N$. Sampling error in the forecast covariance matrix $\widehat{C}_{j+1}$ leads systematically to underestimation of the analysis covariance $C_{j+1}$. In severe cases, this can lead to filter divergence, whereby repeated underestimation of the forecast covariance leads to the filter putting increasingly more weight on the forecasts than observations, eventually becoming unresponsive to observations.

The underestimation of the analysis covariance is often mitigated by inflating the covariance of the forecast ensemble, although it can also be applied to the analysis ensemble instead. The most common form of inflation is *multiplicative inflation*, whereby the forecast ensemble is modified as

$$\widehat{v}_{j+1}^{(n)} \to \widehat{m}_{j+1} + \alpha(\widehat{v}_{j+1}^{(n)} - \widehat{m}_{j+1}), \tag{7.28}$$

with $\alpha \geq 1$ being the inflation parameter. This corresponds to scaling the covariance $\widehat{C}_{j+1}$ by $\alpha^2$.

We provide two illustrative examples motivating the underestimation of the analysis covariance.

**Example 7.13.** Suppose that we assimilate a scalar observation of the $k$th variable into the $i$th variable, i.e., $H$ is a $1 \times d$ vector with a 1 in the $k$th position and zeros elsewhere. It may then be shown that

$$(C)_{ii} - (\widehat{C})_{ii} = \frac{-(\widehat{C})^2_{ik}}{(\widehat{C})_{kk} + (\Gamma)_{kk}}.$$

Now, suppose that variables $i$ and $k$ are in fact uncorrelated, i.e., $(\widehat{C})_{ik} = 0$ when estimated with an infinite ensemble $N \to \infty$. In an EnKF, due to sampling error resulting from a finite ensemble, the estimated $(\widehat{C})_{ik} \neq 0$, leading to a spurious reduction in the analysis covariance, with respect to its value in the predicted ensemble. ♢

**Example 7.14.** Inflation is also used to compensate for an unknown $\Sigma$. If the $\xi_j^{(n)}$ are excluded in (7.23a), then the covariance $\widehat{C}_{j+1}$ will be underestimated in (7.25a). ♢

Theoretical results on the impact of sampling error on EnKFs, including results on the optimal value of the inflation parameter under assumptions on the model dynamics, are referenced in the bibliography, Section 7.12.

### 7.6.3 Localization

In addition to the undersampling effects from the previous subsection, further negative impacts of sampling errors on EnKFs can arise in spatially extended systems. In such physical systems there is typically decay of correlations with distance; sampling error can induce spurious long-range correlations. In the empirical covariance matrices computed by the EnKFs, nearby points in space are likely to be truly correlated, while ones that are farther apart may only have spurious correlations from sampling error. These spurious correlations not only lead to an underestimation of the analysis covariance, as discussed in Example 7.13, but also lead to degraded state estimate due to spurious information transfer between uncorrelated variables.

*Localization* damps covariances in the empirical covariance matrices computed by the EnKF according to distance. A common way of implementing localization is by taking the Hadamard product[1] of the empirical covariance with a localization matrix $L$:

$$\widehat{C}_{j+1} \to L \circ \widehat{C}_{j+1}. \tag{7.29}$$

Typically, $L$ is constructed such that the covariance between between the $i$th and $k$th locations is exponentially damped according to the distance $D_{ki}$ between them, with some characteristic length scale $\ell$:

$$(L)_{ik} = e^{-D_{ik}^2/\ell}. \tag{7.30}$$

[1] Also known as Schur product, this computes the elementwise product of two matrices of the same dimension.

The choice of $\ell$ will depend on the ensemble size, the correlation scales in the system, and the time between analysis steps (since this will control how far information has had time to propagate). Localization can also be implemented in other ways, such as domain localization and observation localization, discussed in the bibliography, Section 7.12; we will restrict our discussion to covariance localization (7.29).

Remark 7.15. If $h(\cdot)$ is linear, the localized $\widehat{C}_{j+1}$ is used to compute the gain as per (7.25b). If $h(\cdot)$ is nonlinear, then localization needs to be applied to both $\widehat{C}^{vh}_{j+1}$ and $\widehat{C}^{hh}_{j+1}$. If the $h(\cdot)$ is such that each observation has a corresponding spatial location, then localization can be applied as discussed above. If some observations correspond to, e.g., spatially integrated quantities, then localization is not straightforward to apply. $\diamondsuit$

Remark 7.16. Besides the sampling error considerations discussed above, localization is also important in increasing the rank of the forecast covariance matrix. When the ensemble size $N$ is less than the system dimension $d$, the forecast covariance will be rank deficient. This implies that the analysis ensemble will lie in the span of the forecast ensemble, and thus that analysis increments are restricted to an $N$-dimensional subspace. Localization typically increases the rank of the forecast covariance, bypassing this subspace problem. $\diamondsuit$

## 7.7 Bootstrap Particle Filter

A more general methodology for approximating the filtering distribution with an interacting ensemble is the particle filter (PF). We make the empirical approximation to the filtering distribution $\pi_j \approx \pi_j^{\mathrm{PF}}$. Here $\pi_0^{\mathrm{PF}} = \pi_0 = \mathcal{N}(m_0, C_0)$ and, for $j \in \mathbb{N}$,

$$\pi_j^{\mathrm{PF}} = \sum_{n=1}^{N} w_j^{(n)} \delta_{\widehat{v}_j^{(n)}}, \tag{7.31}$$

where the particles $\widehat{v}_j^{(n)}$ and weights $w_j^{(n)}$ evolve according to

$$\widehat{v}_{j+1}^{(n)} = \Psi(v_j^{(n)}) + \xi_j^{(n)}, \qquad v_j^{(n)} \overset{\text{i.i.d.}}{\sim} \pi_j^{\mathrm{PF}}, \tag{7.32a}$$

$$\ell_{j+1}^{(n)} = \exp\Big(-\frac{1}{2}\big|y_{j+1}^{\dagger} - h(\widehat{v}_{j+1}^{(n)})\big|_{\Gamma}^{2}\Big), \tag{7.32b}$$

$$w_{j+1}^{(n)} = \ell_{j+1}^{(n)} \Big/ \Big(\sum_{m=1}^{N} \ell_{j+1}^{(m)}\Big). \tag{7.32c}$$

Here $\xi_j^{(n)} \sim \mathcal{N}(0, \Sigma)$ are Gaussian random variables, i.i.d. with respect to both $n$ and $j$.

At each step, the particle filter alternates sampling from the Markovian dynamics, the prediction $\mathsf{P}$, implemented in (7.32a), followed by an application of Bayes theorem, the analysis $\mathsf{A}(\cdot; y_{j+1}^{\dagger})$ which is implemented using importance sampling as in (7.31), with weights as in (7.32c). The resulting algorithm is known as the bootstrap particle filter.

Remark 7.17. The bootstrap particle filter is provably convergent to the true filtering distribution as $N \to \infty$, under quite general conditions, including filters that are far from Gaussian; this should be contrasted with ensemble methods as discussed in Remark 7.12. On the other hand the bootstrap particle filter can perform poorly in high dimensions, suffering from weight collapse. Ensemble methods have been designed to perform well in high dimensions, and methodology for this is also discussed in Remark 7.12. $\diamondsuit$

## 7.8 Optimal Particle Filters

The optimal particle filter differs from the bootstrap particle filter by reversing the order of the Bayesian inference step and the sampling from a Markovian kernel based on the dynamics. In particular, the optimal particle filter first incorporates the observation $y_{j+1}$ by evaluating the likelihood weights $\mathbb{P}(y_{j+1}|v_j)$ for each particle, and then samples from the probability distribution $\mathfrak{p}_{j+1} := \mathbb{P}(v_{j+1}|v_j, y_{j+1})$. This has the potential advantage that the kernel in the optimal particle filter $\mathfrak{p}_{j+1}$ incorporates knowledge of the current observation; in contrast the bootstrap particle filter simply predicts with the unconditioned forecast kernel $\mathsf{P}$, drawing samples from $\mathbb{P}(v_{j+1}|v_j)$ and then reweighting them to reflect the observation.

Remark 7.18. *Optimality* here refers to minimizing the variance of the weights, over one step of the filter, over a wide class of possible particle based methods. Once cycled through multiple steps $j$, this optimality property is lost. However there is a clear intuitive benefit in using the optimal rather than the bootstrap particle filter: in the optimal setting the particles that are reweighted, in one step, use information about the new data point; in the bootstrap setting they do not. $\diamondsuit$

In general nonlinear settings, it may not be possible to implement the optimal particle filter exactly. This arises for two reasons. First, the likelihood weights must integrate the dependence on the latent variable $v_{j+1}$. That is, the integral

$$\mathbb{P}(y_{j+1}|v_j) = \int \mathbb{P}(y_{j+1}|v_{j+1})\, \mathbb{P}(v_{j+1}|v_j)\, dv_{j+1},$$

may not have a closed form. Second, sampling exactly from $\mathfrak{p}_{j+1}$ may not be possible.

One setting where it is tractable to evaluate the likelihood weights and sample from $\mathfrak{p}_{j+1}$ is when the observation model is linear, i.e., $h(\cdot) = H\cdot$ for some $H \in \mathbb{R}^{k\times d}$. Let us consider the dynamic and observation models

$$v_{j+1}^\dagger = \Psi(v_j^\dagger) + \xi_j^\dagger, \tag{7.33a}$$

$$y_{j+1}^\dagger = Hv_{j+1}^\dagger + \eta_{j+1}^\dagger\,. \tag{7.33b}$$

For this dynamics/data model we make the same Gaussian and independence assumptions on the initialization and noise as made in equations (7.1), (7.2) and Assumption 7.3. In this case, the models for propagation of the state and observation can be combined to yield the likelihood function

$$\mathbb{P}(y_{j+1}|v_j) = \mathcal{N}\big(H\Psi(v_j), H\Sigma H^\top + \Gamma\big). \tag{7.34}$$

The Markovian kernel for the dynamics is given by

$$\begin{aligned}\mathbb{P}(v_{j+1}|v_j,y_{j+1}) &\propto \mathbb{P}(y_{j+1}|v_{j+1},v_j)\,\mathbb{P}(v_{j+1}|v_j)\\ &= \mathbb{P}(y_{j+1}|v_{j+1})\,\mathbb{P}(v_{j+1}|v_j).\end{aligned}$$

This density is a quadratic function of $v_{j+1}$ and so following a similar derivation to the update for 3DVar, the Markov kernel is Gaussian, i.e., $\mathbb{P}(v_{j+1}|v_j,y_{j+1}) = \mathcal{N}(m_{j+1},C)$, where the mean $m_{j+1}$ is given by

$$m_{j+1} = (I-KH)\Psi(v_j) + Ky_{j+1} \tag{7.35}$$

and the covariance $C$ satisfies

$$C = (I-KH)\Sigma, \tag{7.36a}$$

$$K = \Sigma H^\top S^{-1}, \tag{7.36b}$$

$$S = H\Sigma H^\top + \Gamma. \tag{7.36c}$$

Given the dynamics and observation model and the observation $y_{j+1}$, we can sample from $\mathfrak{p}_{j+1}$ by sampling from the Gaussian kernel $\mathcal{N}(m_{j+1},C)$, and we can reweight using likelihood (7.34) to build a sample approximation for $\pi_{j+1}$.

To define the OPF algorithm we define matrix triple $(C,K,S)$ by (7.36) and make the empirical approximation to the filtering distribution $\pi_j \approx \pi_j^{\mathrm{OPF}}$. Here $\pi_0^{\mathrm{OPF}} = \pi_0 = \mathcal{N}(m_0,C_0)$ and, for $j \in \mathbb{N}$,

$$\pi_j^{\mathrm{OPF}} = \sum_{n=1}^{N} w_j^{(n)} \delta_{\widehat{v}_j^{(n)}} \tag{7.37}$$

where the particles $\widehat{v}_j^{(n)}$ and weights $w_j^{(n)}$ evolve according to, for $\xi_{n+1}^{(n)}$ i.i.d. $\mathcal{N}(0,C)$,

$$\widehat{v}_{j+1}^{(n)} = (I-KH)\Psi(v_j^{(n)}) + Ky_{j+1}^\dagger + \xi_{j+1}^{(n)}, \qquad v_j^{(n)} \overset{\text{i.i.d.}}{\sim} \pi_j^{\mathrm{OPF}}, \tag{7.38a}$$

$$\ell_{j+1}^{(n)} = \exp\Bigl(-\frac{1}{2}|y_{j+1}^\dagger - H\Psi(v_j^{(n)})|_S^2\Bigr), \tag{7.38b}$$

$$w_{j+1}^{(n)} = \ell_{j+1}^{(n)} / \sum_{m=1}^{N} \ell_{j+1}^{(m)}. \tag{7.38c}$$

## 7.9 4DVar

We introduce the 4DVar methodology, a nomenclature which abbreviates the specific *weak constraint 4DVar* formulation that we focus on here. In short this method computes a MAP estimator (see Definition 1.6) for the smoothing problem from Section 7.2. To this end we define

$$V := \{v_0, \dots, v_J\}$$

and set

$$\mathsf{R}(V) := \frac{1}{2}|v_0 - m_0|_{C_0}^2 + \frac{1}{2}\sum_{j=0}^{J-1} |v_{j+1} - \Psi(v_j)|_\Sigma^2. \tag{7.39}$$

Note that $\exp(-\mathsf{R}(V))$ is proportional to the prior density given in (7.8). We also define

$$\mathsf{L}(V;Y^\dagger) := \frac{1}{2}\sum_{j=0}^{J-1}|y^\dagger_{j+1} - h(v_{j+1})|^2_\Gamma. \tag{7.40}$$

Note that $V \in \mathbb{R}^{d(J+1)}$ and $Y^\dagger \in \mathbb{R}^{kJ}$ and that $\exp(-\mathsf{L}(V;Y^\dagger))$ is proportional to the prior density given in (7.10). We add the expressions in (7.39) and (7.40) to obtain

$$\mathsf{J}(V;Y^\dagger) = \mathsf{R}(V) + \mathsf{L}(V;Y^\dagger), \tag{7.41}$$

noting that $\exp(-\mathsf{J}(V))$ is proportional to the posterior density given in (7.11). The MAP estimator for this posterior is given by

$$V^\star \in \arg\min_{V\in\mathbb{R}^{d(J+1)}} \mathsf{J}(V;Y^\dagger). \tag{7.42}$$

We then take $V^\star$ as our estimate of $V^\dagger$.

Remark 7.19. Minimization (7.42) is often implemented using Newton or Gauss–Newton methodologies (see Section 16.4). Ensemble approximations of Gauss-Newton are also used.

The problem (7.42) is the (weak constraint) 4DVar method. Strong constraint 4DVar is found by letting $\Sigma \to 0$, resulting in a minimization problem over $v_0 \in \mathbb{R}^d$. The name 4DVar refers to the fact that, in weather forecasting, the optimization problem is for a field in three physical space dimensions and one time dimension.

As with all MAP estimators, the result of applying 4DVar is simply a point estimator. To obtain information about the posterior with density proportional to $\exp(-\mathsf{J}(V))$ requires the use of methods such as Markov chain Monte Carlo (MCMC) and sequential Monte Carlo (SMC), or variational approximations. ♢

## 7.10 Reanalysis

*Reanalysis*, also known as retrospective analysis, is concerned with retrospectively producing an estimate of the trajectory of a system, or the conditional probability distribution of the states given the observations. There may be several reasons for doing this: (i) since filtering is often done in an online mode, returning to find a smoothing estimate of the past uses more information from the observations; (ii) it may be desirable to have an estimate of the past which is actually a trajectory of a proposed dynamics model, something not provided by filtering; (iii) new observations of the past may become available; (iv) the model, or filtering/smoothing method of choice may change over time. A reanalysis dataset uses the same model and data assimilation method throughout the entire interval, and incorporates all possible observations. Reanalyses can be produced by either filters or smoothers, although the latter is preferred since they make maximum use of the available observations.

Remark 7.20. An important feature of reanalysis is that it consolidates possibly sparse, irregular observations to give at every time step a complete state estimate of the system on the model state space. The extent to which the information from observed parts of the system can transfer to unobserved parts is formalized under the concept of *observability*; see the Bibliography for references. Moreover, reanalysis datasets, due to the impact of the observational nudging, are often assumed to have lower impact of model error (see Section 7.11) than the raw forecasts produced by the model. This is discussed in Section 8.4. For these two reasons, reanalyses are often used to train ML forecast models. ♢

## 7.11 Model Error

We now consider the data assimilation problem in the setting where parts of the model are unknown, building on discussions in Section 1.4 for the general inverse problem. To describe *forecast model error*, we introduce parameter $\vartheta$ which captures unknown aspects of the systematic part of the dynamical model. The stochastic dynamics model becomes

$$v^\dagger_{j+1} = \Psi_\vartheta(v^\dagger_j) + \xi^\dagger_j,\ j \in \mathbb{Z}^+, \tag{7.43a}$$

$$v^\dagger_0 \sim \mathcal{N}(m_0, C_0),\ \xi^\dagger_j \sim \mathcal{N}(0, \Sigma) \text{ i.i.d.}, \tag{7.43b}$$

where we assume that sequence $\{\xi^\dagger_j\}$ is independent of initial condition $v^\dagger_0$; this is often written as $\{\xi^\dagger_j\} \perp\!\!\!\perp v^\dagger_0$. To describe *observation model error* we introduce parameter $\varphi$ which captures unknown aspects of the observation operator. The data model becomes

$$y^\dagger_{j+1} = h_\varphi(v^\dagger_{j+1}) + \eta^\dagger_{j+1},\ j \in \mathbb{Z}^+, \tag{7.44a}$$

$$\eta^\dagger_j \sim \mathcal{N}(0, \Gamma) \text{ i.i.d.}, \tag{7.44b}$$

where we assume that $\{\eta^\dagger_j\} \perp\!\!\!\perp v^\dagger_0$ for all $j$ and $\eta^\dagger_k \perp\!\!\!\perp \xi^\dagger_j$ for all $j, k$.

Remark 7.21. We define unknown parameter $\theta = (\vartheta, \Sigma, \varphi, \Gamma)$. For simplicity we assume we learn all components of $\theta$. However we recognize that it will often be of interest to learn only a subset of the parameters. Furthermore, the covariance matrices $\Sigma$ and $\Gamma$ may themselves be parameterized (for example as unknown parameter multiplying the identity). ♢

In the following two subsections we formulate model error in the smoothing and filtering contexts. Chapter 8 is devoted to detailed algorithms concerned with learning model error.

### 7.11.1 Model Error: Smoothing

To formulate learning of model error in the context of a smoothing problem we put a prior on $(V^\dagger, \theta)$, defined by the stochastic dynamics model for $V^\dagger|\theta$ and prior on $\theta$, and we condition on a likelihood defined by the data model. Building on Section 7.2 we define prior

$$\rho(V^\dagger, \theta) \propto \exp\Big(-\frac{1}{2}|v^\dagger_0 - m_0|^2_{C_0} - \frac{1}{2}\sum_{j=0}^{J-1} |v^\dagger_{j+1} - \Psi_\vartheta(v^\dagger_j)|^2_\Sigma\Big)\rho_\theta(\theta), \tag{7.45}$$

where we have written the prior on $V^\dagger|\theta$ and then multiplied by prior $\rho_\theta(\theta)$ on $\theta$. As in Section 7.2 we define

$$\eta^\dagger := \{\eta_1^\dagger, \dots, \eta_J^\dagger\},$$

noting that $\eta^\dagger \sim \mathcal{N}(0, \mathsf{\Gamma})$, where $\mathsf{\Gamma}$ is block diagonal with $\Gamma$ in each diagonal block. If we then define

$$\mathsf{h}_\varphi(V^\dagger) := \{h_\varphi(v_1^\dagger), \dots, h_\varphi(v_J^\dagger)\}$$

then the data model may be written as

$$Y^\dagger = \mathsf{h}_\varphi(V^\dagger) + \eta^\dagger.$$

We are interested in finding $\mathbb{P}(V^\dagger, \theta, |Y^\dagger) = \Pi^{Y^\dagger}(V^\dagger, \theta)$. We may apply Bayes Theorem 1.2, with prior $\rho$ noting that, under this prior, $V^\dagger \perp\!\!\!\perp \eta^\dagger$, the standard setting for Bayesian inversion from Chapter 1. We obtain

$$\Pi^{Y^\dagger}(V^\dagger, \theta) \propto \exp\Big(-\frac{1}{2}|Y^\dagger - \mathsf{h}_\varphi(V^\dagger)|_{\mathsf{\Gamma}}^2\Big)\rho(V^\dagger, \theta). \tag{7.46}$$

### 7.11.2 Model Error: Filtering

It is also possible to use techniques based on filtering. To illustrate this idea we consider the setting in which $\Sigma$ and $\Gamma$ are known and the unknown $\theta$ comprise only $\vartheta$ and $\varphi$ appearing in $\Psi_\vartheta$ and $h_\varphi$, respectively. Consider the following dynamical system, holding for all $j \in \mathbb{Z}^+$ :

$$v_{j+1}^\dagger = \Psi_\vartheta(v_j^\dagger) + \xi_j^\dagger, \tag{7.47a}$$
$$\vartheta_{j+1}^\dagger = \vartheta_j^\dagger, \tag{7.47b}$$
$$\varphi_{j+1}^\dagger = \varphi_j^\dagger, \tag{7.47c}$$
$$y_{j+1}^\dagger = h_\varphi(v_{j+1}^\dagger) + \eta_{j+1}^\dagger. \tag{7.47d}$$

This can now be viewed as a filtering problem for $\{(v_j^\dagger, \vartheta_j^\dagger, \varphi_j^\dagger)\}$ given the data generated by $\{y_j^\dagger\}$. As it stands it is slightly out of the scope of problems studied in Section 7.1, because no noise appears in the evolution of the unknown parameters. However, as mentioned in Remark 7.4, filtering can be extended to this setting. Alternatively it is possible to replace equations (7.47b),(7.47c) by stochastic processes and to use an average over index $j$ of the filtering solution to provide a parameter estimate; in this context autoregressive Gaussian processes are natural.

### 7.11.3 Filtering with Small Model Error

Assume that the true dynamics are given by systematic component defined by function $\Psi$. If we learn an approximation $\Psi_\vartheta$ of the form (8.15) it will inevitably make errors in representing $\Psi$, either because of lack of expressivity of the function class $\Psi_\vartheta^{\text{correction}}$ and/or because of finite available data. Ideally, data assimilation algorithms will not only filter the noise but also compensate for errors in approximating $\Psi$. The next theorem provides the conditions under which a 3DVar filter reduces model error as data is acquired sequentially, on average. In particular, we show that the error in the reconstructed solution will scale proportionally to the noise in the observations and the model error.

**Assumption 7.22.** *Consider the dynamics and data model (7.1), (7.2), assume that the dynamics is noise-free ($\Sigma = 0$) and that the observation operator is linear ($h(\cdot) = H\cdot$). Let the sequence of observations $y^\dagger_{j+1}$ be found from observing a true signal $v^\dagger_j$ given by*

$$\begin{aligned} v^\dagger_{j+1} &= \Psi(v^\dagger_j), \\ y^\dagger_{j+1} &= Hv^\dagger_{j+1} + \eta^\dagger_{j+1}. \end{aligned}$$

**Theorem 7.23.** *Let Assumption 7.22 hold and let $K$ be a gain matrix appearing in the mean update of the 3DVar method with an approximate forecast model $\Psi^a$:*

$$m_{j+1} = (I - KH)\Psi^a(m_j) + Ky^\dagger_{j+1}.$$

*Assume that $\Psi^a$ is close to $\Psi$ in the sense that, for some $\delta \in [0, \infty)$,*

$$\sup_{v\in\mathbb{R}^d} \big|(I - KH)\big(\Psi(v) - \Psi^a(v)\big)\big| = \delta.$$

*Assume, further, the observability condition that there exists constant $\lambda \in (0, 1)$ so that*

$$\sup_{v\in\mathbb{R}^d} |(I - KH)D\Psi(v)| \le \lambda,$$

*and denote*

$$\epsilon = \mathbb{E}|K\eta_j|.$$

*Then, the 3DVar estimate based on the biased forecast model satisfies*

$$\limsup_{j\to\infty} \mathbb{E}|m_j - v^\dagger_j| = \frac{\epsilon + \delta}{1 - \lambda}.$$

*Proof.* The mean update in 3DVar with the biased forecast model is given by

$$m_{j+1} = (I - KH)(\Psi(m_j) + b_j) + Ky^\dagger_{j+1},$$

where

$$b_j := \Psi^a(m_j) - \Psi(m_j)$$

represents bias in the forecast model at time $j$. Using the source of the observations, the error relative to the dynamics for the true signal $v^\dagger_{j+1}$ is given by

$$\begin{aligned} m_{j+1} - v^\dagger_{j+1} &= (I - KH)(\Psi(m_j) + b_j) + K(H\Psi(v^\dagger_j) + \eta^\dagger_{j+1}) - \Psi(v^\dagger_j) \\ &= (I - KH)(\Psi(m_j) - \Psi(v^\dagger_j)) + K\eta_{j+1} + (I - KH)b_j \\ &= \int_0^1 (I - KH)D\Psi(sm_j + (1 - s)v^\dagger_j)(m_j - v^\dagger_j)\, ds + K\eta_{j+1} + (I - KH)b_j, \end{aligned}$$

where in the last line we used the mean value theorem for $\Psi$ on the line segment $sm_j + (1-s)v_j^\dagger$ for $s \in [0,1]$. By the triangle inequality, the error is bounded by

$$\begin{aligned}|m_{j+1} - v_{j+1}^\dagger| &\le \Big(\int_0^1 |(I-KH)D\Psi(sm_j + (1-s)v_j^\dagger)|\, ds\Big)|m_j - v_j^\dagger| + |K\eta_{j+1}| + \delta \\ &\le \Big(\int_0^1 \lambda\, ds\Big)|m_j - v_j^\dagger| + |K\eta_{j+1}| + \delta.\end{aligned}$$

Taking an expectation over the measurement errors, we have

$$\mathbb{E}|m_{j+1} - v_{j+1}^\dagger| \le \lambda \mathbb{E}|m_j - v_j^\dagger| + \epsilon + \delta.$$

Letting $e_j \coloneqq \mathbb{E}|m_{j+1} - v_{j+1}^\dagger|$ and applying the discrete Gronwall's inequality we have

$$e_j \le \lambda^j e_0 + (\epsilon + \delta)\frac{1-\lambda^j}{1-\lambda}.$$

Noting that $\lambda < 1$, then $\lambda^j \to 0$ as $j \to \infty$ and so the result follows. □

Remark 7.24. Theorem 7.23 highlights that running the 3DVar algorithm for long times can correct for small model error. Moreover, this model error only needs to be controlled in the unobserved directions. That is, the overall error of the recovered state will be small if $(I-KH)b_j$ is small for all $j$, rather than the overall model error $b_j$ being small. Ideally, the gain $K$ in 3DVar is chosen such that the conditions in the assumption of Theorem 7.23 hold; see the bibliography Section 8.5 for one such guarantee under an observability condition; this assumption could be checked when learning the correction. ♢

## 7.12 Bibliography

For overviews of the subjects of data assimilation and filtering/smoothing, see the text books [163, 188, 269, 15, 280, 19, 67, 91] and the review paper [268]. Furthermore the books [168, 206] and the review paper [48] comprise pedagogical introductions to data assimilation in the context of weather forecasting, turbulence modeling, and geophysical sciences, respectively. The importance of data assimilation for numerical weather forecasting was articulated in [242], then known as *objective analysis*, and connected to sequential estimation theory in [109].

The Kalman filter was introduced in [167]. Discussion of the steady state covariance, and results concerning convergence to the steady state, may be found in [181]. The ideas of 3DVar for the solution of the analysis step was introduced in the context of weather forecasting in [201]. For definition of the (cycled) 3DVar algorithm as employed here, see [188]. For analysis of the (cycled) 3DVar algorithm see [187, 222].

The EnKF was introduced in [90]. See [15, 91, 188, 269] for more recent discussion of the methodology. The methodology based around optimization has a probabilistic interpretation in the Gaussian setting, and this can be used to justify the empirical approximation (7.27). In the Gaussian setting see [190, 208]; in the near Gaussian setting see [50]. For an introduction to the particle filter see [82]. For the optimal particle filter see [81]. Particle filters typically suffer from weight collapse in high dimensional problems: all weights become zero, except one [28, 291].

The impact of sampling error on EnKFs in the case that the signal is a Gaussian process is addressed in [272]. Localization and inflation are reviewed in [91]. Insights into localization are given in [3]. There are many variants of the EnKF, including ensemble square-root filters that are more computationally suited for high-dimensional problems than the formulation presented in this chapter [312, 155]. Since these variants avoid the construction of the covariance matrices, localization is often implemented in these filters through *domain localization*, where the spatial domain is divided into multiple areas, and analyses done locally in each [155].

We note that besides the considerations about spurious correlations and forecast covariance rank discussed in the chapter, localization can also be interpreted in a dynamical systems context. For an EnKF, one generally needs enough ensemble members to span the unstable–neutral subspace, corresponding to the number of non-negative Lyapunov exponents, in order to prevent filter divergence. This observation is supported by the fact that, in the case of linear dynamics, the forecast covariance matrix will collapse onto the unstable–neutral subspace; see the review of data assimilation for chaotic dynamics [49] for references to these results. It has been observed, however, that dynamical systems such as the atmosphere are locally low dimensional [245, 236]; that is, the dynamics within a spatial region may have a significantly lower dimension (as quantified by the dimension of the subspace spanned by the fastest growing modes) than that of the entire system, implying that filtering may be successful in such cases when localization is applied.

For a discussion of 4DVar, in the context of weather forecasting, see [98]. As with all MAP estimators, the result of applying 4DVar is simply a point estimator. To obtain information about the posterior with density proportional to $\exp\big(-\mathsf{J}(V)\big)$ requires the use of methods such as MCMC [41], SMC [76] or variational methods [165].

The concept of reanalysis originates in atmospheric science [317]. Although operational weather forecasting has been done for decades, there have been considerable changes in forecast models and data assimilation methods, necessitating reanalysis to have a consistent trajectory estimate over decadal timescales. One of the first reanalysis datasets for the atmosphere was produced by the National Centers for Environmental Prediction (NCEP) and the National Center for Atmospheric Research (NCAR) [169]. The subject of observability is reviewed in [216].

Model error in data assimilation is often modelled by Gaussian noise, and encapsulated in the model noises $\xi^\dagger$. Deterministic formulations of model error are reviewed in [46]. The special case of model bias is considered in [74]. Time-correlated model error is considered in [7]. Methods for correcting model error are discussed in Chapter 8, and a bibliography provided there.

# Chapter 8

## Learning Forecast and Observation Model

In Chapter 7 we considered the data assimilation problem of estimating a time-evolving state in the setting where the models governing the state-observation system are known. It is natural to ask how to extend this to settings where the dynamical and observation models are unknown. This chapter is devoted to the problem of estimating a time-evolving state, together with the dynamics governing its evolution and the data model from a window of partial and noisy observations of the state.

Jointly estimating the state and parameters of the mathematical models governing the dynamics and data is important in data assimilation applications where available stochastic models are inaccurate or expensive to evaluate. The accuracy of data assimilation can be improved, for instance, by estimating parameters in the model dynamics or by learning a neural network model error correction; the efficiency of data assimilation can be improved by leveraging machine-learned surrogate models that are cheap to evaluate.

This chapter describes algorithmic frameworks to blend data assimilation techniques for estimating the state with machine learning techniques for recovering model parameters. After introducing the problem setting in Section 8.1, Section 8.2 describes the expectation-maximization (EM) framework for joint parameter and state estimation. In particular, we will present practical algorithmic instances of this framework to estimate parameters defining the deterministic mappings $\Psi$ and $h$ in the dynamics-data model and/or defining the model error covariances within the dynamics-data model. Section 8.3 discusses auto-differentiable Kalman filters, an approach to maximum likelihood estimation of the parameters that relies on Kalman-type algorithms to approximate the likelihood function, and on auto-differentiation to approximate its gradient. Section 8.4 considers the specific problem of learning model error corrections for the dynamics. Section 8.5 closes with extensions and bibliographical remarks. The material in this chapter brings together many topics covered in these notes, including the variational formulations of Bayes theorem (Chapter 2), ensemble Kalman and particle filtering algorithms (Chapter 7), parameterizations of functions for machine learning (Chapter 14), and optimization (Chapter 16).

## 8.1 The Setting

Consider the stochastic dynamics and observation models given by

$$
\begin{aligned}
v_{j+1} &= \Psi_\vartheta(v_j) + \xi_j, && \xi_j \sim \mathcal{N}(0,\Sigma) \text{ i.i.d.}, && \text{(8.1a)}\\
y_{j+1} &= h_\varphi(v_{j+1}) + \eta_{j+1}, && \eta_j \sim \mathcal{N}(0,\Gamma) \text{ i.i.d.}, && \text{(8.1b)}
\end{aligned}
$$

with $v_0 \sim \mathcal{N}(m_0, C_0)$ and $v_0 \perp \{\xi_j\} \perp \{\eta_j\}$ for $0 \le j \le J-1$. We assume that $v_j \in \mathbb{R}^d$ and $y_j \in \mathbb{R}^k$. We denote by $\theta \coloneqq \{\vartheta, \varphi, \Sigma, \Gamma\}$ the collection of unknown model parameters in the deterministic maps $\Psi_\vartheta$, $h_\varphi$ and the error covariances. We could extend to parameterizing the error covariances, as discussed in Remark 7.21.

The stochastic dynamics in (8.1) defines the Markov kernel $\mathbb{P}(\cdot\,|v_j,\theta) \coloneqq \mathcal{N}\big(\Psi_\vartheta(v_j),\Sigma\big)$ and the data model in (8.1) defines the likelihood function for the observations $\mathbb{P}(\cdot\,|v_{j+1},\theta) \coloneqq \mathcal{N}\big(h_\varphi(v_{j+1}),\Gamma\big)$. Our goal is to find $\theta$ so that the Markov kernel approximates that of an unknown, true, stochastic dynamics model, and the likelihood approximates the true data-generating process, given together in the form

$$
\begin{aligned}
v^\dagger_{j+1} &= \Psi(v^\dagger_j) + \xi_j, && \xi_j \sim \mathcal{N}(0,\Sigma^\dagger), \qquad 0 \le j \le J-1, && \text{(8.2)}\\
y^\dagger_{j+1} &= h(v^\dagger_{j+1}) + \eta_{j+1} && \eta_j \sim \mathcal{N}(0,\Gamma^\dagger). && \text{(8.3)}
\end{aligned}
$$

Once $\theta$ is found, the signal $\{v^\dagger_j\}$ can be estimated leveraging the filtering and smoothing algorithms in Chapter 7 of these notes using the models in (8.1).

For a given and fixed integer $J$, we define, following the notation in Chapter 7,

$$
V^\dagger \coloneqq \{v^\dagger_0, \ldots, v^\dagger_J\}, \qquad Y^\dagger \coloneqq \{y^\dagger_1, \ldots, y^\dagger_J\}.
$$

Note that $V^\dagger \in \mathbb{R}^{d(J+1)}$ and $Y^\dagger \in \mathbb{R}^{kJ}$. In contrast to the supervised learning setting in Chapter 14, here we do not assume to have direct data $\{v^{(n)}, \Psi(v^{(n)})\}_{n=1}^N$ to learn the map $\Psi$. Instead, we assume to have only access to data $Y^\dagger$ obtained from indirect and noisy measurement of a trajectory $V^\dagger$ from the signal.

**Data Assumption 8.1.** *The data available is $Y^\dagger \coloneqq \{y^\dagger_1, \ldots, y^\dagger_J\}$ a single draw from the marginal $\kappa$ on the observed data.*

Before presenting algorithms to recover the parameter $\theta$, we consider three motivating examples for the parameter $\vartheta$, of the parameterized map $\Psi_\vartheta$, and of the relationship between the map $\Psi_\vartheta$ and the ground-truth deterministic map $\Psi$.

**Example 8.2** (Parameterized Dynamics)**.** $\Psi = \Psi_{\vartheta^\dagger}$ is parameterized, but the true parameter $\vartheta^\dagger$ is unknown and needs to be estimated. For instance, $\Psi$ may be defined by the time-discretization of a parameterized system of differential equations and we may be interested in estimating $\vartheta^\dagger$ from observations $Y^\dagger$. ♢

**Example 8.3** (Fully Unknown Dynamics)**.** There are settings in which $\Psi$ is fully unknown and $\vartheta$ can represent the parameters of, for example, a neural network, random features, or Gaussian process surrogate model $\Psi_\vartheta$ for $\Psi$; see Chapter 14 for background on these parameterizations of functions. The goal is to find an accurate surrogate model $\Psi_\vartheta$. Even if $\Psi$ is known, a surrogate model, or *emulator*, may be cheaper to evaluate, enabling the use of large sample sizes for particle filters or ensemble Kalman methods; see Chapter 11. Similar computational considerations motivate learning the forward map for inverse problems in Chapter 3. $\diamondsuit$

**Example 8.4** (Model Correction). $\Psi$ is unknown, but we have access to an inaccurate model $\Psi^a \approx \Psi$. Here $\vartheta$ can represent, for example, the parameters of a neural network, random features, or Gaussian process $\Psi_\vartheta^{\text{correction}}$ used to correct the inaccurate model $\Psi^a$. The goal is to learn $\vartheta$ so that $\Psi_\vartheta := \Psi^a + \Psi_\vartheta^{\text{correction}}$ approximates $\Psi$ accurately. Learning model corrections is important in applications where available models have moderate predictive accuracy. For instance, fine scales of the state may not be resolved accurately due to computational constraints and we may be interested in learning a surrogate model from data which accounts for the unresolved scales of the system. $\diamondsuit$

## 8.2 Expectation Maximization

In contrast to the supervised learning setting in Chapter 14, here we do not assume to have direct data $\{v^{(n)}, \Psi(v^{(n)})\}_{n=1}^N$ to learn the map $\Psi$. Instead, we assume to have only access to data $Y^\dagger$ obtained from indirect and noisy measurement of a signal trajectory $V$: see Data Assumption 8.1. To address the learning problem we view it as a *missing data* problem in which ideally we estimate $\theta$ from $V$; but the data $V$ is missing and we must first estimate it from $Y^\dagger$. To frame this idea we adopt a *maximum likelihood* approach to estimate $\theta$. That is, we seek to find $\theta$ that maximizes the marginal likelihood of $\theta$ given the observed data $Y^\dagger$

$$\mathbb{P}(Y^\dagger|\theta) = \int \mathbb{P}(Y^\dagger, V|\theta)\, dV = \int \mathbb{P}(Y^\dagger|V,\theta)\, \mathbb{P}(V|\theta)\, dV. \tag{8.4}$$

The likelihood in the last equation marginalizes over a distribution $\mathbb{P}(V|\theta)$ for the unknown states $V$; here and throughout this chapter integrals are over $\mathbb{R}^{d(J+1)}$ unless otherwise noted. Evaluating the likelihood function $\mathbb{P}(Y^\dagger|\theta)$ is challenging due to the unobserved, or *latent*, variable $V$, which depends on the unknown parameters $\theta$.

Remark 8.5. Likelihood functions that are computationally expensive to evaluate, such as the integral in (8.4), motivated the introduction of likelihood-free methods for Bayesian inference in Chapter 6. In this chapter, we will focus on optimization algorithms, rather than Bayesian methods, to find point estimators for the parameter $\theta$. $\diamondsuit$

Since the likelihood in (8.4) involves an expectation over a distribution depending on the unknown parameters, it is natural to consider an iterative process for finding the maximum likelihood estimator of $\theta$. One widely-adopted iterative procedure is the *expectation-maximization* (EM) algorithm, which alternates between an expectation, (E) step, related to obtaining a lower bound for the integral (8.4), and a maximization (M) step, related to updating the parameter estimate by maximizing the lower bound.

Theorem 16.10 and the non-negativity of the KL divergence imply that, for any distribution $q(V)$ for the unknown state $V$, it holds that

$$\log \mathbb{P}(Y^\dagger|\theta) \geq \mathcal{L}(q,\theta) := \int \log\left(\frac{\mathbb{P}(Y^\dagger, V|\theta)}{q(V)}\right) q(V)\, dV.$$

Given the current iterate $\theta^\ell$ of the algorithm, in the E step we choose $q^\ell(V) := \mathbb{P}(V|Y^\dagger, \theta^\ell)$ to obtain a lower bound $\mathcal{L}(q^\ell, \theta)$ for $\log \mathbb{P}(Y^\dagger|\theta)$. It follows from Theorem 16.10 that the choice $q^\ell(V) = \mathbb{P}(V|Y^\dagger, \theta^\ell)$ solves the optimization problem

$$q^\ell \in \arg\max_q \mathcal{L}(q, \theta^\ell).$$

Theorem 16.10 further implies that, at $\theta = \theta^\ell$, equality is achieved: $\log \mathbb{P}(Y^\dagger|\theta^\ell) = \mathcal{L}(q^\ell, \theta^\ell)$. The lower bound $\mathcal{L}(q^\ell, \theta)$ is practical to compute because the expectation is with respect to a known distribution for the unknown state variables, and the joint distribution $\mathbb{P}(Y^\dagger, V|\theta)$ is tractable from the parameterized stochastic dynamics and observation models; see (8.5) below. We refer the reader to Section 16.3 for more background on the EM algorithm.

The M-step defines the updated parameters $\theta^{\ell+1}$ by maximizing the lower bound $\mathcal{L}(q^\ell, \theta)$ with respect to $\theta$. That is,

$$\theta^{\ell+1} \in \arg\max_\theta \mathcal{L}(q^\ell, \theta) = \arg\max_\theta \mathbb{E}^{V\sim q^\ell}\big[\log \mathbb{P}(Y^\dagger, V|\theta)\big].$$

From the dynamics and observation models in (8.1), the joint distribution of $V$ and $Y^\dagger$ admits the characterization

$$\begin{aligned}\log \mathbb{P}(Y^\dagger, V|\theta) &= \log \mathbb{P}(Y^\dagger|V, \theta) + \log \mathbb{P}(V|\theta)\\ &= -\frac{1}{2}\log\det(\Gamma) - \frac{1}{2}\sum_{j=0}^{J-1}|y_{j+1} - h_\varphi(v_{j+1})|_\Gamma^2\\ &\quad -\frac{1}{2}|v_0 - m_0|_{C_0}^2 - \frac{1}{2}\log\det(\Sigma) - \frac{1}{2}\sum_{j=0}^{J-1}|v_{j+1} - \Psi_\vartheta(v_j)|_\Sigma^2 + c,\end{aligned} \tag{8.5}$$

where $c$ is a constant independent of $V$, $Y^\dagger$ and $\theta$. We recall that the E-step integrates the log-likelihood in (8.5) with respect to $q^\ell(V) = \mathbb{P}(V|Y^\dagger, \theta^\ell)$. The updated parameters in the M-step can then be computed by defining and maximizing the loss function

$$\mathsf{L}(\theta; q^\ell) = -\frac{1}{2}\log\det(\Gamma) - \frac{1}{2}\int \sum_{j=0}^{J-1}|y_{j+1} - h_\varphi(v_{j+1})|_\Gamma^2 q^\ell(V)\, dV \tag{8.6}$$

$$\qquad - \frac{1}{2}\log\det(\Sigma) - \frac{1}{2}\int\sum_{j=0}^{J-1}|v_{j+1} - \Psi_\vartheta(v_j)|_\Sigma^2 q^\ell(V)\, dV,$$

$$\theta^{\ell+1} \in \arg\max_\theta \mathsf{L}(\theta; q^\ell). \tag{8.7}$$

To illustrate the application of the EM algorithm, the next subsections introduce practical algorithmic frameworks to define empirical loss functions and update the parameters. Subsection 8.2.1 focuses on estimating the dynamics and observation model error covariances. The following two subsections focus on estimating the parameters in the dynamics model assuming the observation model is known; Subsection 8.2.2 uses a particle approximation of $\mathbb{P}(V|Y^\dagger, \theta^\ell)$ in (8.6), while Subsection 8.3 uses Gaussian approximations for $\mathbb{P}(V|Y^\dagger, \theta^\ell)$ that are derived from an extended (or ensemble) Kalman method.

### 8.2.1 Learning Observation and Model Error Covariances

In this subsection we consider the setting where the dynamics operator $\Psi$ and observation operator $h$ are known, but the dynamics and observation error covariances are unknown. That is, we seek the unknown parameters $\theta = \{\Gamma, \Sigma\}$. In this setting, we notice that the loss function (8.6) in the M-step is separable as the sum of two loss functions that each depend on only one parameter $\Gamma$ or $\Sigma$. Furthermore, both of the two loss functions depending on $\Gamma$ and $\Sigma$ respectively have the same form as the objective in Proposition 2.9 for an unknown covariance in a Gaussian variational inference problem. Thus, the loss can be explicitly maximized over $\Gamma, \Sigma$. Setting $\theta^\ell = \{\Gamma_\ell, \Sigma_\ell\}$, and using the explicit optimizer, we find the covariance matrix updates

$$\Gamma^{\ell+1} = \int \sum_{j=0}^{J-1} \Big(y_{j+1} - h(v_{j+1})\Big) \otimes \Big(y_j - h(v_{j+1})\Big)\, q^\ell(V)\, dV, \tag{8.8a}$$

$$\Sigma^{\ell+1} = \int \sum_{j=0}^{J-1} \Big(v_{j+1} - \Psi(v_j)\Big) \otimes \Big(v_{j+1} - \Psi(v_j)\Big)\, q^\ell(V)\, dV. \tag{8.8b}$$

The following result shows that this update for the parameters $\Gamma, \Sigma$ leads to a monotonic increase of the log-likelihood for the observed variables $Y^\dagger$.

**Theorem 8.6.** *The iterates (8.8) of the EM algorithm for $\{\Gamma, \Sigma\}$ satisfy*

$$\log \mathbb{P}(Y^\dagger|\theta^{\ell+1}) \geq \log \mathbb{P}(Y^\dagger|\theta^\ell).$$

*Proof.* The loss function $\mathsf{L}(\theta; q^\ell)$ in the M-step is convex in $\theta$ and so $\theta^{\ell+1} = \{\Gamma^{\ell+1}, \Sigma^{\ell+1}\}$ are the unique global optimum parameters, i.e., $\theta^{\ell+1} \in \arg\max_\theta \mathsf{L}(\theta; q^\ell)$. From Theorem 16.11, the iterates of the EM algorithm with the exact optimal parameters in the M-step satisfies the monotonic increase in the log-likelihood. □

Remark 8.7. The monotonic increase of the log likelihood in the population setting does not guarantee that the EM iterations converge. Additional assumptions are required to show that there is a strict improvement in the log-likelihood at each iteration and to provide a condition where the parameters reach a local critical point. Furthermore, in practice, we will have samples from an empirical approximation of $\mathbb{P}(V|Y^\dagger, \theta^\ell)$ and so we may update the covariances using Monte Carlo approximations of the covariance matrices in (8.8). The added randomness from Monte Carlo approximation introduces a potentially different approximation of $\mathbb{P}(V|Y^\dagger, \theta^\ell)$ in each of the E-steps; consequently, the monotonic increase in the log-likelihood is no longer guaranteed. ♢

### 8.2.2 Monte Carlo EM

In this subsection we consider a more general setting of learning both the dynamics model and forecast error covariance. We will focus on estimating parameters in the dynamics model assuming the observation model is known. That is, the unknown parameters correspond to $\theta = \{\vartheta, \Sigma\}$. Given (approximate) samples $\{v^{(n)}\}_{n=1}^{N}$ from $q^{\ell}(V) = \mathbb{P}(V|Y^{\dagger}, \theta^{\ell})$, we may use Monte Carlo to approximate the expectation in the loss function as

$$\mathsf{L}(\theta; q^{\ell}) \approx c + -\frac{1}{2}\log\det(\Sigma) - \frac{1}{2N}\sum_{n=1}^{N}\sum_{j=0}^{J-1}|v_{j+1}^{(n)} - \Psi_{\vartheta}(v_j^{(n)})|_{\Sigma}^{2}, \tag{8.9}$$

where $c$ accounts for the first two terms in (8.6), which are constant with respect to the unknown parameters $\theta$ as defined in this subsection. As in the last subsection, we notice that, given $\vartheta$, the expression (8.9) can be explicitly maximized over $\Sigma$. Indeed, the optimal solution is given by the covariance matrix

$$\Sigma^{\ell+1} = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=0}^{J-1}\Big(v_{j+1}^{(n)} - \Psi_{\vartheta}(v_j^{(n)})\Big) \otimes \Big(v_{j+1}^{(n)} - \Psi_{\vartheta}(v_j^{(n)})\Big).$$

We note that this covariance depends on the parameters $\vartheta$ via the model $\Psi_{\vartheta}$. To maximize the loss $\mathsf{L}(\theta; q^{\ell})$ over $\theta = \{\vartheta, \Sigma\}$, we can employ iterative optimization methods: alternate optimization over $\Sigma$, for fixed $\vartheta$, as given in the preceding block, with optimization over $\vartheta$. Algorithm 8.1 summarizes a resulting EM type algorithm, for learning dynamics and states, using Monte Carlo and gradient ascent updates. We notice that this implementation of the EM framework treats differently the parameters $\vartheta$ and $\Sigma$ in the M-step to account for the closed-form update for $\Sigma^{\ell+1}$.

The distribution $\mathbb{P}(V|Y^{\dagger}, \theta^{\ell})$ for the signal $\{v_j\}$ can be estimated by performing data assimilation using the stochastic dynamics and data models in (8.1) with learned parameter $\vartheta^{\ell}$ and covariance model $\Sigma^{\ell}$. These samples may be obtained, for instance, using Markov chain Monte Carlo methods. Alternately, the ensemble Kalman methods described in Chapter 7 may be employed, with the proviso that they can yield accurate posterior samples only in approximately linear-Gaussian settings.

Remark 8.8. In analogy to the supervised learning task studied in Chapter 14, equation (8.6) may be conceptually interpreted as defining a risk for the parameter $\theta$, where the distribution $\mathbb{P}(V|Y^{\dagger}, \theta^{\ell})$ represents our available knowledge of the latent variable $V$ given data $Y^{\dagger}$ and parameter estimate $\theta^{\ell}$. Then, (8.9) can be interpreted as an empirical risk, defined via approximate samples $V^{(n)} \sim \mathbb{P}(V|Y^{\dagger}, \theta^{\ell})$. ♢

Remark 8.9. While we present Algorithm 8.1 using gradient ascent, in practice we may employ other optimization methods such as accelerated first-order methods or second-order methods, like Gauss-Newton to optimize the parameter $\vartheta$; we refer to Chapter 16 for more details. The choice of algorithm may depend on computational constraints and properties of the loss function, such as its smoothness. ♢

**Algorithm 8.1** Expectation-Maximization with Monte Carlo and Gradient Ascent

```
**Input**: Initialization $\theta^0 = \{\vartheta^0, \Sigma^0\}$. Rule for gradient ascent step-sizes $\{\alpha^i\}_{i=1}^{\mathcal{I}}$.
For $\ell = 0, 1, \ldots, L-1$ do the following expectation and maximization steps:
**E-Step**: Obtain (approximate) samples $\{V^{(n)}\}_{n=1}^N$ from $\mathbb{P}(V|Y^\dagger, \theta^\ell)$ and use these samples to approximate $\mathsf{L}(\theta; q^\ell)$ as in (8.9).
**M-Step**:
Set
    $\Sigma^{\ell+1}(\vartheta) = \frac{1}{2N}\sum_{n=1}^N \sum_{j=0}^{J-1} \Bigl(v_{j+1}^{(n)} - \Psi_\vartheta(v_j^{(n)})\Bigr) \otimes \Bigl(v_j^{(n)} - \Psi_\vartheta(v_j^{(n)})\Bigr)$
and define
    $\mathsf{J}\bigl(\vartheta, \{V^{(n)}\}_{n=1}^N\bigr) := -\frac{1}{2N}\sum_{n=1}^N \sum_{j=0}^{J-1} \Bigl|v_{j+1}^{(n)} - \Psi_\vartheta(v_j^{(n)})\Bigr|_{\Sigma^{\ell+1}(\vartheta)}^2.$
Initialize $\vartheta^{\ell,0} = \vartheta^\ell$.
**for** $i = 0, \ldots, \mathcal{I}-1$ **do**
    $\vartheta^{\ell,i+1} = \vartheta^{\ell,i} + \alpha^i D_\vartheta \mathsf{J}(\vartheta^{\ell,i}, \{V^{(n)}\}_{n=1}^N)$.
**end for**
Set $\vartheta^{\ell+1} = \vartheta^{\ell,\mathcal{I}}$.
**Output**: Approximation $\theta^L$ to the maximum likelihood estimator.
```

## 8.3 Auto-Differentiable Kalman Filters

As in the previous section we focus on estimating parameters in the dynamics model assuming that the observation model is known, so that the unknown parameters correspond to $\theta = \{\vartheta, \Sigma\}$. Unlike the previous subsection, here we assume that the observation operator is linear so that $h(v) = Hv$. for some matrix $H \in \mathbb{R}^{k \times d}$. We again work under Data Assumption 8.1.

In Section 7.5 we studied extended Kalman filtering (ExKF) algorithms that make a Gaussian approximation to the prediction and analysis distributions. Specifically

$$\mathbb{P}(v_{j+1}|Y_j^\dagger, \theta) \approx \mathcal{N}(\widehat{m}_{j+1}(\theta), \widehat{C}_{j+1}(\theta)) \tag{8.10}$$

for the predictive distribution; and

$$\mathbb{P}(v_{j+1}|Y_{j+1}^\dagger, \theta) \approx \mathcal{N}(m_{j+1}(\theta), C_{j+1}(\theta)) \tag{8.11}$$

for the analysis distribution. These ExKF-based Gaussian approximations are accurate if the noise in the dynamics and the observations are small, or the map $\Psi_\vartheta$ is approximately linear; see Remark 7.10. They may also be used to justify similar Gaussian approximations computed from the ensemble Kalman filter; in particular the ensemble means and covariance provide Gaussian ansatzs for the predictive and analysis distributions. However, we concentrate here on the ExKF.

In this section we show how the Gaussian ansatz (8.10) can be leveraged to produce a ExKF-based approximation of the log-likelihood function $\log \mathbb{P}(Y^\dagger|\theta)$ and thereby an algorithm for maximum likelihood estimation of parameters $\theta$ in the stochastic dynamics model. It is intrinsic to the methodology presented that the observation operator is linear. When combined with the Gaussian ansatz this enables exact integration over the state, in order to find the marginal distribution on $Y^\dagger$ given $\theta$. The derivation rests on the following characterization of the log-likelihood function under a Gaussian ansatz.

**Theorem 8.10.** *Assume that the observation operator is linear: $h(\cdot) = H\,\cdot$ . Suppose that, for each $0 \le j \le J-1$, the predictive distribution $\mathbb{P}(v_{j+1}|Y_j^\dagger, \theta)$ of the stochastic dynamics and data models (8.1) is Gaussian with mean $\widehat{m}_{j+1}(\theta)$ and covariance $\widehat{C}_{j+1}(\theta)$. Then the log-likelihood function admits the following characterization*

$$\log \mathbb{P}(Y^\dagger|\theta) = -\frac{1}{2}\sum_{j=0}^{J-1} \bigl|y_{j+1} - H\widehat{m}_{j+1}(\theta)\bigr|^2_{S_{j+1}(\theta)} - \frac{1}{2}\sum_{j=0}^{J-1} \log\det\bigl(S_{j+1}(\theta)\bigr), \tag{8.12}$$

*where $S_{j+1}(\theta) = H\widehat{C}_{j+1}(\theta)H^\top + \Gamma$.*

*Proof.* We have

$$\log \mathbb{P}(Y^\dagger|\theta) = \sum_{j=0}^{J-1} \log \mathbb{P}(y_{j+1}|Y_j^\dagger, \theta),$$

where we use the convention that $Y_0^\dagger := \emptyset$ so that conditioning to $Y_0^\dagger$ does not provide any information. Now conditioning in the data model

$$y_{j+1} = Hv_{j+1} + \eta_{j+1},$$

we see that

$$\begin{aligned}
\mathbb{E}\bigl[y_{j+1}|Y_j^\dagger, \theta\bigr] &= \mathbb{E}\bigl[Hv_{j+1} + \eta_{j+1}|Y_j^\dagger, \theta\bigr] = H\widehat{m}_{j+1}(\theta),\\
\mathrm{Cov}\bigl[y_{j+1}|Y_j^\dagger, \theta\bigr] &= \mathrm{Cov}\bigl[Hv_{j+1} + \eta_{j+1}|Y_j^\dagger, \theta\bigr] = H\widehat{C}_{j+1}(\theta)H^\top + \Gamma.
\end{aligned}$$

Moreover, $\mathbb{P}(y_{j+1}|Y_j^\dagger, \theta)$ is Gaussian. The result follows. □

Thus, for any value of $\theta$, we may obtain an *approximation* of the log-likelihood $\log \mathbb{P}(Y^\dagger|\theta)$ by running a Kalman filtering algorithm, obtaining predictive means and covariances $\bigl(\widehat{m}_{j+1}(\theta), \widehat{C}_{j+1}(\theta)\bigr)$, for $0 \le j \le J-1$, and using (8.12). Moreover, an approximation of the log-likelihood gradient at $\theta$ can be obtained auto-differentiating through our Kalman-based likelihood estimate. Auto-differentiable Kalman filters use these estimates of the log-likelihood gradients to conduct gradient ascent. The procedure is summarized in Algorithm 8.2, for which we define

$$\mathsf{J}(\theta) := -\frac{1}{2}\sum_{j=0}^{J-1} \bigl|y_{j+1} - H\widehat{m}_{j+1}(\theta)\bigr|^2_{S_{j+1}(\theta)} - \frac{1}{2}\sum_{j=0}^{J-1} \log\det\bigl(S_{j+1}(\theta)\bigr). \tag{8.13}$$

**Algorithm 8.2** Auto-Differentiable Kalman Filter

```
Input: Initialization $\theta^0$, rule to choose step-sizes $\{\alpha^\ell\}_{\ell=0}^{L-1}$.
For $\ell = 0, 1, \dots, L-1$ do the following Kalman filtering and gradient ascent steps:
    Kalman Filtering: Run an extended (or ensemble) Kalman filtering algorithm to obtain predictive means and covariances $\widehat{m}_{j+1}(\theta^\ell), \widehat{C}_{j+1}(\theta^\ell)$, for $0 \le j \le J-1$.
    Gradient Ascent: Auto-differentiate the map $\theta \mapsto \mathsf{J}(\theta)$ defined in (8.13) to obtain a gradient estimate $D\mathsf{J}(\theta^\ell)$. Set
        $$\theta^{\ell+1} = \theta^\ell + \alpha^\ell D\mathsf{J}(\theta^\ell). \tag{8.14}$$
Output: Approximation $\theta^L$ to the maximum likelihood estimator.
```

Remark 8.11. If desired, the signal $\{v_j\}$ can be estimated by performing data assimilation with the stochastic dynamics and data models (8.1) with learned parameter $\theta^L$. ♢

Remark 8.12. Notice that in the EM Algorithm 8.1 each posterior sample $\{V^{(n)}\}_{n=1}^N$ is used to perform $\mathcal{I}$ gradient descent steps in the M-Step, with the goal of maximizing the lower bound $\mathcal{L}(q^\ell, \theta)$. In contrast, in the auto-differentiable Kalman filter Algorithm 8.2, each run of a Kalman filtering algorithm is used to produce an estimate of the log-likelihood gradient, and a gradient ascent step is taken. This is possible because we can analytically integrate out the state variable, given a linear observation operator and Gaussian predictive distribution. ♢

## 8.4 Correcting Model Error Using Analysis Increments

We again work under Data Assumption 8.1. However we now use the learned trajectories $\{v_j^{(n)}\}_{j=0}^{J-1}$ obtained from an ensemble method to create surrogate supervised learning problems.

Here we consider the problem of learning parameters $\theta = \{\vartheta\}$ in a correction term to the systematic component of the stochastic dynamics model (8.1a). To this end we assume that

$$\Psi_\vartheta = \Psi^a + \Psi_\vartheta^{\text{correction}}. \tag{8.15}$$

Thus, $\Psi^a$ is a known approximation to the systematic component of the stochastic dynamics model; we wish to use data to find a parameterized correction $\Psi_\vartheta^{\text{correction}}$.

For simplicity we assume the covariance for the dynamical model error is known and we will work in the Monte Carlo setting of Subsection 8.2.2. In this case, the loss function in (8.6) has the form

$$\mathsf{L}(\theta; q^\ell) = c - \frac{1}{2N} \sum_{n=1}^{N} \sum_{j=0}^{J-1} \bigl| v_{j+1}^{(n)} - \Psi^a(v_j^{(n)}) - \Psi_\vartheta^{\text{correction}}(v_j^{(n)}) \bigr|_\Sigma^2, \tag{8.16}$$

where $c$ is a constant that is independent of the parameters to be estimated. We recognize that minimizing (8.16) over $\theta$ corresponds to the supervised learning problem of estimating the mapping from $v_j$ to the increments $v_{j+1} - \Psi^a(v_j)$ given (approximate) analysis samples $V^{(n)}$.

Remark 8.13. The Monte Carlo samples, indexed by $n$, of the time-series $\{v_j^{(n)}\}_{j=0}^{J-1}$, may have been created without the model error correction, i.e., assuming $\Psi^a$ is the true dynamics model, in which case the loss in (8.16) is simply $\mathsf{L}(\theta)$. Or they may have been created from the corrected model for a setting $\vartheta_\ell$ that was found in an earlier iteration of the EM algorithm. In the latter case, the maximization of $\mathsf{L}(\cdot; q^\ell)$ yields $\theta^{\ell+1}$. ♢

## 8.5 Discussion and Bibliography

In this chapter we have introduced two computational frameworks for joint state and parameter estimation: the EM algorithm and auto-differentiable filters. Both frameworks find the parameters in the dynamics by approximating the maximum likelihood estimator, and then use a filtering or smoothing algorithm to recover the state. This final section provides bibliographical context for the algorithms considered in this chapter, and briefly discusses other computational methods that do not stem from a maximum likelihood formulation.

Embedding of the EnKF and the ensemble Kalman smoother (EnKS) into the EM algorithm was proposed in [307, 315, 83, 257], with a focus on estimation of error covariance matrices. The E-step is approximated with an EnKS under the Monte Carlo EM framework [331]. In addition, [39, 233] incorporate machine learning techniques in the M-step to train neural network surrogate models. The paper [33] proposes Bayesian estimation of model error statistics, together with a neural network emulator for the dynamics. On the other hand, [316, 308, 63] consider online EM methods for error covariance estimation with EnKF. Online methods aim to reduce computation by not reprocessing the smoothing distribution for each new observation. Although gradient information is used during the M-step to train the surrogate models in [39, 233, 33], these methods do not auto-differentiate through the EnKF.

Our presentation of auto-differentiable Kalman filters follows [57], which proposes and analyzes an approach for joint state and parameter estimation that leverages gradient information of an EnKF estimate of the likelihood. EnKFs for derivative-free maximum likelihood estimation are studied in [298, 257]. An empirical comparison of the likelihood computed using the EnKF and other filtering algorithms is made in [47]; see also [129, 219]. The paper [84] uses EnKF likelihood estimates to design a pseudo-marginal MCMC method for Bayesian inference of model parameters. The works [297, 299] propose online Bayesian parameter estimation using the likelihood computed from the EnKF under a certain family of conjugate distributions.

While our discussion has focused on ensemble Kalman methods, particle filters can also be employed for joint state and parameter estimation. Particle filters give an unbiased estimate of the data likelihood [75, 12]. Based upon this likelihood estimate, a particle MCMC Bayesian parameter estimation method is designed in [12]. Although particle filter likelihood estimates are unbiased, they suffer from two potential drawbacks that limit their applicability to some problems. First, their variance can be large, as they inherit the weight degeneracy of importance sampling in high dimensions

[291, 2, 275, 276]—see Chapter 7 for further background on this subject. Second, while the forecast and analysis steps of particle filters can be auto-differentiated, the resampling steps involve discrete distributions that cannot be handled by the reparameterization trick, as discussed in [57]. For this reason, previous differentiable particle filters omit auto-differentiation of the resampling step [227, 205, 189], introducing a bias.

An alternative to maximum likelihood estimation is to optimize a lower bound of the data log-likelihood with variational inference [30, 173, 263]. The posterior distribution over the latent states is approximated with a parametric distribution and is jointly optimized with model parameters defining the dynamics model. In this direction, variational sequential Monte Carlo methods [227, 205, 189] construct the lower bound using a particle filter. Moreover, the proposal distribution of the particle filter is parameterized and jointly optimized with model parameters. Although variational sequential Monte Carlo methods provide consistent data log-likelihood estimates, they suffer from the same two potential drawbacks as likelihood-based particle filter methods. A recent work [161] proposes blending variational sequential Monte Carlo and EnKF with an importance sampling-type lower bound estimate, which is effective if the state dimension is small. Other works that build on the variational inference framework include [176, 264, 99, 209]. An important challenge is to obtain suitable parameterizations of the posterior, especially when the state dimension is high. For this reason, a restrictive Gaussian parameterization with a diagonal covariance matrix is often used in practice [176, 99]. The topic of variational approximations to filtering and smoothing problems is discussed in Chapter 9.

Another alternative approach to maximum likelihood estimation is to concatenate state and parameters into an augmented state-space, and employ the data assimilation methods in Part II of these notes in the augmented state-space. This approach requires one to design a pseudo-dynamic for the parameters, which can be challenging when certain types of parameters (e.g., error covariance matrices) are involved [297, 77] or if the dimension of the parameters is high. The use of EnKF for joint learning of state and model parameters by *state augmentation* was introduced in [9].

Beyond the problem of joint parameter and state estimation, the development of data-driven frameworks for learning dynamical systems is a very active research area. We refer to [194] for a framework and to [44, 119, 132, 261] for recent methods that do not rely on the EM algorithm, auto-differentiation of filtering methods, or variational inference.

In addition, data-driven machine learning frameworks have gained popularity for combining physics-based models with model error corrections. These hybrid models may avoid avoid the over-smoothing or unphysical behavior of predictive models that replace the entire forecast dynamics [35]. Two common representations for model error include additive corrections [93, 36] and data-driven subgrid scale parameterizations [265, 34]; we refer to Section 1.4 for other forms of model error. We note that the approach discussed in this chapter learns the corrections offline, as online methods typically require the adjoint operator of the physics-based model, which may be challenging to obtain for some physics-based models, e.g., in numerical weather prediction and for climate modeling [200].

While the effect of model bias on data assimilation is an open research topic, [74] analyzed Kalman filtering in the presence of model bias. Under observability conditions, [216] showed that the Kalman gain $K$ can be designed to satisfy the condition in (**??**) for stability of a data assimilation scheme with linear dynamics; this is also related to the design of Luenberger observers in control theory.

# Chapter 9

## Learning Parameterized Filters and Smoothers

In this chapter we use learning to solve the filtering and smoothing problems introduced in Chapter 7. We consider both state estimation, finding a representative sequence of states from the data and knowledge of the model, and probabilistic estimation, approximating the posterior distributions for sequences of states given data. In Section 9.1 we will first introduce the variational formulation for smoothing and filtering problems, which enable variational inference approaches to solve these problems, in analogy to the procedures in Chapter 2 for inverse problems. Sections 9.2 and 9.3 introduce practical instances of the variational formulation for recovering a smoothed point estimator of the time-varying state and an approximation to the smoothing distribution within a tractable family of distributions, respectively. We refer to the approaches in Sections 9.2 and 9.3 as state estimation and probabilistic estimation, respectively, to emphasize the computed uncertainty in the latter approach. Sections 9.4 and 9.5 introduce analogous approaches of state estimation and probabilistic estimation for filtering, respectively. The learned approximations we consider in this chapter are built as generalizations of the classic filtering and smoothing algorithms presented in Chapter 7.

## 9.1 Variational Formulations of Smoothing and Filtering

In this section we describe how to formulate the probabilistic approaches to both smoothing and filtering using the variational framework introduced in Chapter 2. In particular, we seek the smoothing distribution as the solution to an optimization problem in Subsection 9.1.1 and we show how to find the filtering distribution in Subsection 9.1.2. In these subsections we propose variational inference approaches to derive tractable computational methods to solve the proposed variational smoothing and filtering problems. We define the notation $V = \{v_0, \dots, v_J\}$ and $V_j = \{v_0, \dots, v_j\}$, and recall $Y^\dagger = \{y_1^\dagger, \dots, y_J^\dagger\}$ and $Y_j^\dagger := \{y_1^\dagger, \dots, y_j^\dagger\}$ as in (7.3). Throughout most of this section we make the following assumption about data available for learning:

**Data Assumption 9.1.** *The data available is $Y^\dagger := \{y_1^\dagger, \dots, y_J^\dagger\}$ a single draw from the marginal $\kappa$ on the observed data.*

On one occasion we go beyond this setting, in this section, making Data Assumption 9.4. It is convenient to use the notation $\mathcal{P}^J := \mathcal{P}(\mathbb{R}^{d(J+1)})$.

### 9.1.1 Variational Formulation of Smoothing

The density for the smoothing distribution $\mathbb{P}(V|Y^\dagger)$, following Section 7.2, is then

$$\Pi^{Y^\dagger}(V) \propto \mathsf{l}(Y^\dagger|V)\rho(V),$$

where the likelihood and the prior are given by

$$\mathsf{l}(Y^\dagger|V) = \prod_{j=1}^{J} \mathsf{l}_j(v_j), \tag{9.1a}$$

$$\mathsf{l}_j(v_j) = Z_\Gamma^{-1}\exp(-\frac{1}{2}|y_j^\dagger - h(v_j)|_\Gamma^2), \quad Z_\Gamma = (2\pi)^{k/2}\det(\Gamma)^{1/2}, \tag{9.1b}$$

$$\rho(V) = \prod_{j=0}^{J} \rho_j(v_j), \tag{9.1c}$$

$$\rho_j(v_j) = \begin{cases} Z_\Sigma^{-1}\exp(-\frac{1}{2}|v_j - \Psi(v_{j-1})|_\Sigma^2) & j \geq 1, \quad Z_\Sigma = (2\pi)^{d/2}\det(\Sigma)^{1/2}, \\ Z_{C_0}^{-1}\exp(-\frac{1}{2}|v_0 - m_0|_{C_0}^2) & j = 0, \quad Z_{C_0} = (2\pi)^{d/2}\det(C_0)^{1/2}. \end{cases} \tag{9.1d}$$

The following theorem shows that the smoothing density arises as the solution of a variational problem. This result is analogous to the variational formulation for the posterior in an inverse problem, that was presented in Theorem 2.2.

**Theorem 9.2.** *Consider the likelihood $\mathsf{l}(Y^\dagger|V)$ and prior $\rho(V)$ given in (9.1), and let*

$$\mathsf{J}(q) = \mathsf{D}_{\mathrm{KL}}(q\|\rho) - \mathbb{E}^q[\log \mathsf{l}(Y^\dagger|V)], \tag{9.2a}$$

$$q_{\mathrm{OPT}} \in \arg\min_{q\in\mathcal{P}^J} \mathsf{J}(q). \tag{9.2b}$$

*Then, the minimizer of $\mathsf{J}$ over $\mathcal{P}^J$ is the smoothing density, i.e., $q_{\mathrm{OPT}} = \Pi^{Y^\dagger}$.*

*Proof.* This follows directly from the variational formulation of Bayes Theorem 2.2. □

This formulation of the smoothing problem can be used as the basis for algorithms to learn approximations for the smoothing distribution from data.

### 9.1.2 Variational Formulation of Filtering

The goal in this subsection is to find variational formulations of the filtering problem that are useful as the basis of algorithms to learn from data. We present two different variational formulations of filtering: one that involves a variational inference problem at every analysis time, and one that begins from the above variational formulation of the smoothing problem and imposes the temporal structure of filtering on it.

#### Variational Inference at Each Assimilation Step

Recall the prediction ($\mathsf{P}$) and analysis ($\mathsf{A}(\cdot; y_{j+1}^\dagger)$) maps that define filtering (7.4). The analysis step (7.7) is an application of Bayes Theorem 1.2 with likelihood $\mathbb{P}(y_{j+1}^\dagger|v_{j+1})$ and prior $\widehat{\pi}_{j+1} = \mathbb{P}(v_{j+1}|Y_j^\dagger)$ found by application of $\mathsf{P}$ given by (7.6) to $\pi_j$. Thus, the analysis density $\pi_{j+1}$ can be written as the solution to an optimization problem using the variational formulation of Bayes Theorem 2.2:

$$\mathsf{J}_{j+1}(q) = \mathsf{D}_{\mathrm{KL}}(q\|\widehat{\pi}_{j+1}) - \mathbb{E}^{v_{j+1}\sim q}\big[\log \mathbb{P}(y^{\dagger}_{j+1}|v_{j+1})\big], \tag{9.3a}$$

$$\pi_{j+1} \in \arg\min_{q\in\mathcal{P}} \mathsf{J}(q), \tag{9.3b}$$

where $\mathcal{P} := \mathcal{P}(\mathbb{R}^d)$. The above assumes that we have access to the true prior $\widehat{\pi}_{j+1}$. We now consider densities $q'_j$ recursively defined by the prediction–analysis cycle

$$q'_0 = \pi_0, \tag{9.4a}$$

$$\mathsf{J}_{j+1}(q) = \mathsf{D}_{\mathrm{KL}}(q\|\mathsf{P}q'_j) - \mathbb{E}^{v_{j+1}\sim q}\big[\log \mathbb{P}(y^{\dagger}_{j+1}|v_{j+1})\big], \tag{9.4b}$$

$$q'_{j+1} \in \arg\min_{q\in\mathcal{P}} \mathsf{J}_{j+1}(q). \tag{9.4c}$$

Then, $q'_{j+1}$ equals the filtering distribution $\mathbb{P}(v_{j+1}|Y^{\dagger}_{j+1})$. Performing the minimization (9.4c) over a restricted class of probability densities, potentially defined differently at each step $j$, leads to a sequence of approximate filtering densities $\{q'_j\}_{j\geq 0}$.

### Imposing Filtering Structure on Smoothing

Alternatively, one can start from smoothing and then impose a temporal structure on the resulting density. In filtering, the probability distribution for state $v_j$ at time $j$ only depends on past observations $Y^{\dagger}_j$, as defined in (7.3), and not future observations. In order to approximate a filtering distribution it is natural to seek minimizers of $\mathsf{J}(\cdot)$ defined by (9.2a) which factorize in the form

$$q(V) = \prod_{j=1}^{J} \mathfrak{q}_j(v_j|V_{j-1};Y^{\dagger}_j)\mathfrak{q}_0(v_0). \tag{9.5}$$

For later use we also define

$$q_j(V_j) = \prod_{i=1}^{j} \mathfrak{q}_i(v_i|V_{i-1};Y^{\dagger}_i)\mathfrak{q}_0(v_0). \tag{9.6}$$

We let $\mathcal{C}^J \subset \mathcal{P}^J$ denote probability density functions of the form (9.5). (The use of $\mathcal{C}$ is to invoke the conditional, auto-regressive structure.) Ideally we would like to minimize the loss function $\mathsf{J}(\cdot)$ defined in (9.2a) over $\mathcal{C}^J$. However there is no mechanism in loss function (9.2a) to impose the specific data dependence inherent in $\mathcal{C}^J$, namely that $q_j$ depends only on $Y^{\dagger}_j$, within the context of minimizing $\mathsf{J}(\cdot)$; this is because only one instance of the data $Y^{\dagger}$ is available. We now describe two possible ways of circumventing this issue. The first involves minimization of a sequence of objective functions $\mathsf{J}_j(\cdot)$ each of which only sees $Y^{\dagger}_j$. The second involves use of an objective function $\mathsf{J}(\cdot)$ defined over multiple realizations of $Y^{\dagger}$.

**Multiple Objective Functions** The methodology suggested here is based on the following theorem, which employs the notation (9.5) (9.6) and in which it is important to note the distinction between $\mathfrak{q}_j$ and $q_j$:

**Theorem 9.3.** *For $q \in \mathcal{C}^J$ satisfying the factorized structure (9.5), the objective function (9.2) can be written as*

$$\mathsf{J}_j(\mathfrak{q}_j) = \mathsf{D}_{\mathrm{KL}}\Big(\mathfrak{q}_j(v_j|V_{j-1};Y_j^\dagger)\|\rho_j(v_j)\Big) - \mathbb{E}^{v_j\sim\mathfrak{q}_j}\big[\log \mathsf{l}_j(v_j)\big], \tag{9.7a}$$

$$\mathsf{J}(q) = \sum_{j=1}^{J} \mathbb{E}^{V_{j-1}\sim q_{j-1}}\big[\mathsf{J}_j(\mathfrak{q}_j)\big]. \tag{9.7b}$$

*Proof.* We will use the $\mathsf{ELBO}$ introduced in Definition 2.13 and the relationship between $\mathsf{ELBO}$, KL divergence, and likelihood established in Theorem 16.10. We start by noting that

$$\mathsf{D}_{\mathrm{KL}}(q\|\Pi^{Y^\dagger}) = \log \mathbb{P}(Y^\dagger) - \mathsf{ELBO}(\mathbb{P}(V,Y^\dagger),q).$$

Since the first term does not depend on $q$, we focus on the second term, given by

$$\mathsf{ELBO}(\mathbb{P}(V,Y^\dagger),q) = \mathbb{E}^{V\sim q}\left[\log\frac{\mathbb{P}(V,Y^\dagger)}{q(V)}\right].$$

The joint density $\mathbb{P}(V,Y^\dagger)$ can be factorized autoregressively as

$$\mathbb{P}(V,Y^\dagger) = \prod_{j=1}^{J}\mathbb{P}(y_j^\dagger|Y_{j-1}^\dagger,V_j)\mathbb{P}(v_j|Y_{j-1}^\dagger,V_{j-1}).$$

Using this, as well as the factorized structure (9.5) of $q$, we obtain

$$\begin{aligned}\mathsf{ELBO}(\mathbb{P}(V,Y^\dagger),q) &= \mathbb{E}^{q}\left[\log\left(\prod_{j=1}^{J}\frac{\mathbb{P}(y_j^\dagger|Y_{j-1}^\dagger,V_j)\mathbb{P}(v_j|Y_{j-1}^\dagger,V_{j-1})}{\mathfrak{q}_j(v_j|V_{j-1};Y_j^\dagger)}\right)\right]\\ &= \sum_{j=1}^{J}\mathbb{E}^{q}\left[\log\left(\frac{\mathbb{P}(y_j^\dagger|Y_{j-1}^\dagger,V_j)\mathbb{P}(v_j|Y_{j-1}^\dagger,V_{j-1})}{\mathfrak{q}_j(v_j|V_{j-1};Y_j^\dagger)}\right)\right].\end{aligned}$$

Since each term is only conditioned on variables up to time $j$, we can simplify to

$$\begin{aligned}\mathsf{ELBO}(\mathbb{P}(V,Y^\dagger),q) &= \sum_{j=1}^{J}\mathbb{E}^{q_j}\left[\log\left(\frac{\mathbb{P}(y_j^\dagger|Y_{j-1}^\dagger,V_j)\mathbb{P}(v_j|Y_{j-1}^\dagger,V_{j-1})}{\mathfrak{q}_j(v_j|V_{j-1};Y_j^\dagger)}\right)\right]\\ &= \sum_{j=1}^{J}\mathbb{E}^{q_{j-1}}\mathbb{E}^{\mathfrak{q}_j}\left[\log\left(\frac{\mathbb{P}(y_j^\dagger|Y_{j-1}^\dagger,V_j)\mathbb{P}(v_j|Y_{j-1}^\dagger,V_{j-1})}{\mathfrak{q}_j(v_j|V_{j-1};Y_j^\dagger)}\right)\right]\\ &= -\sum_{j=1}^{J}\mathbb{E}^{V_{j-1}\sim q_{j-1}}\Big[\mathsf{D}_{\mathrm{KL}}\Big(\mathfrak{q}_j(v_j|V_{j-1};Y_j^\dagger)\|\rho_j(v_j)\Big) - \mathbb{E}^{v_j\sim\mathfrak{q}_j}\big[\log \mathsf{l}_j(v_j)\big]\Big],\end{aligned}$$

where the last line follows from noticing that $\mathbb{P}(y_j^\dagger|Y_{j-1}^\dagger, V_j) = \mathbb{P}(y_j^\dagger|v_j) = \mathsf{l}_j$ is the observation likelihood at time $j$, and $\mathbb{P}(v_j|Y_{j-1}^\dagger, V_{j-1}) = \mathbb{P}(v_j|v_{j-1}) = \rho_j(v_j)$ is the prior for time $j$. Then, minimizing $\mathsf{D}_{\mathrm{KL}}(q\|\Pi^{Y^\dagger})$ is equivalent to minimizing $\mathsf{J}(q)$ given in (9.7). □

This is an interesting result, but direct minimization of $\mathsf{J}(\cdot)$ as in (9.7b) will not lead to a minimizer in class $\mathcal{C}^J$. This is because minimization will lead to dependence on the entire $Y^\dagger$ in each of the conditionals $\mathfrak{q}_j(v_j|V_{j-1};Y^\dagger)$; there is no mechanism to ensure dependence in the form $\mathfrak{q}_j(v_j|V_{j-1};Y_j^\dagger)$. However, given a single realization of the data $Y^\dagger$ it is reasonable to proceed as follows to enforce this dependence: instead of minimizing $\mathsf{J}(\cdot)$ from (9.7b), we may instead sequentially minimize $\mathsf{J}_j(\cdot)$ from (9.7a), over index $j = 1, \dots, J$. Thus

$$\mathfrak{q}_j^\star \in \arg\min_{\mathfrak{q}\in\mathcal{P}(\mathbb{R}^d)} \mathsf{J}_j(\mathfrak{q}), \tag{9.8a}$$

$$\mathfrak{q}^\star(V) = \prod_{j=1}^{J} \mathfrak{q}_j^\star(v_j|V_{j-1};Y_j^\dagger)\mathfrak{q}_0^\star(v_0). \tag{9.8b}$$

Note that each $\mathsf{J}_j$ depends only on $Y_j^\dagger$ hence enforcing the desired dependence on each conditional $\mathfrak{q}_j^\star(v_j|V_{j-1};Y_j^\dagger)$.

#### Multiple Data Sequences

**Data Assumption 9.4.** *The data available is $\{(Y^\dagger)^{(n)}\}_{n=1}^N$ which are multiple independent realizations from the marginal $\kappa$ on the observed data $Y^\dagger := \{y_1^\dagger, \dots, y_J^\dagger\}$.*

Assuming that we have access to the distribution $\kappa$ on $Y^\dagger$ implied by the stochastic dynamics model (7.1), (7.2), we may consider the following optimization problem:

$$\begin{aligned} &\mathsf{J}(q) = \mathbb{E}^{Y^\dagger}\Big[\mathsf{D}_{\mathrm{KL}}(q\|\rho) - \mathbb{E}^q\big[\log \mathsf{l}(Y^\dagger|V)\big]\Big],\\ &q_{\mathrm{OPT}} \in \arg\min_{q\in\mathcal{C}^J} \mathsf{J}(q). \end{aligned}$$

In practice, expectation over $Y^\dagger$ is approximated empirically using Data Assumption 9.4. Recall that the idea of averaging over multiple data instances was also used in Section 4.5.

## 9.2 Smoothing: State Estimation

Recall the 4DVar approach to the smoothing problem defined in Section 7.9. It finds an estimate of the state $V^\star := \{v_0^\star, \dots, v_J^\star\}$ from the observed data $Y^\dagger := \{y_1^\dagger, \dots, y_J^\dagger\}$. Emphasizing the dependence of $V^\star$ on the available data $Y^\dagger$ we have

$$V^\star(Y^\dagger) \in \arg\min_{V\in\mathbb{R}^{d(J+1)}} \mathsf{J}(V;Y^\dagger), \tag{9.9}$$

$$\mathsf{J}(V;Y^\dagger) = \mathsf{R}(V) + \mathsf{L}(V;Y^\dagger), \tag{9.10}$$

where $\mathsf{R}(V)$ and $\mathsf{L}(V;Y^\dagger)$ are defined in (7.39) and (7.40). Of course $V^\star(Y^\dagger)$ is simply the MAP estimator defined by the Bayesian smoothing problem from Section 7.2.

In the following we assume that we have multiple solutions of the 4DVar MAP estimator for multiple realizations of the data $Y^\dagger$. We use the same notational conventions as in the preceding section.

**Data Assumption 9.5.** *The data available is $\{(Y^\dagger)^{(n)}\}_{n=1}^N$ which are multiple independent realizations from the marginal $\kappa$ on the observed data $Y^\dagger := \{y_1^\dagger, \dots, y_J^\dagger\}$ . And, for each $n$, we have $(V^\star)^{(n)} = V^\star((Y^\dagger)^{(n)})$.*

We employ distance-like deterministic scoring rules $\mathsf{D}$; see Definition 12.56 from Subsection 12.3.7. Given the data $\{(Y^\dagger)^{(n)}, (V^\star)^{(n)}\}_{n=1}^N$ from Data Assumption 9.5 we may apply the methodology of Section 6.4 to learn the mapping from $Y^\dagger$ to $V^\star$. Indeed we find $V^\star(Y^\dagger) \approx V(Y^\dagger;\theta^\star)$ where $\theta^\star$ solves an empirical approximation of the minimization problem

$$\mathsf{J}(\theta) = \mathbb{E}^{Y^\dagger\sim\kappa}\Big[\mathsf{D}\big(V(Y^\dagger;\theta), V^\star(Y^\dagger)\big)\Big], \tag{9.11a}$$

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{J}(\theta). \tag{9.11b}$$

Remark 9.6. It is also possible to circumvent the use of 4DVar and directly generate data pairs $\{(Y^\dagger)^{(n)}, (V^\dagger)^{(n)}\}_{n=1}^N \sim \gamma$, where $\gamma$ denotes the joint distribution on signal and data defined by (7.1), (7.2), (7.3). It is then possible to solve an empirical approximation of

$$\mathsf{J}(\theta) = \mathbb{E}^{(Y^\dagger,V^\dagger)\sim\gamma}\Big[\mathsf{D}\big(V(Y^\dagger;\theta), V^\dagger\big)\Big], \tag{9.12a}$$

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{J}(\theta). \tag{9.12b}$$

♢

## 9.3 Smoothing: Probabilistic Estimation

In this section we discuss how the smoothing distribution may be learned.

### 9.3.1 Smoothing: Gaussian Approximate Probabilistic Estimation

We may apply the ideas of Chapter 2 to the smoothing problem, working under Data Assumption 9.1. We introduce a parametric family of probability density functions $\mathcal{Q} \subset \mathcal{P}^J$ and minimize the objective function from (9.2a) over $\mathcal{Q}$ instead of $\mathcal{P}^J$ :

$$\mathsf{J}(q) = \mathsf{D}_{\mathrm{KL}}(q\|\rho) - \mathbb{E}^q\big[\log \mathsf{l}(Y^\dagger|V)\big], \tag{9.13a}$$

$$q^\star \in \arg\min_{q\in\mathcal{Q}} \mathsf{J}(q). \tag{9.13b}$$

Recall the 4DVar solution (7.42) which is the MAP estimator for the smoothing problem, as introduced in Section 7.9. Precompute this solution of an optimization problem, and call it $V^\star$. Now let

$$\mathcal{Q} = \Big\{q \in \mathcal{P}^J : q = \mathcal{N}(V^\star, \Sigma),\ \Sigma \in \mathbb{R}^{d(J+1)\times d(J+1)}_{\mathrm{sym},>}\Big\}.$$

We thus seek to optimize over parameter $\Sigma$. In practice we may wish to (Cholesky) square-root factorize $\Sigma$, as in Example 5.6, to impose symmetry.

### 9.3.2 Smoothing: Amortized Gaussian Approximate Probabilistic Estimation

We may also be interested in learning a parametric model to capture dependence of an approximate smoothing distribution on the data $Y^\dagger$. To this end, in this subsection we work under working under Data Assumption 9.4. We generalize the previous subsection and choose to minimize over $\mathcal{S}_\theta : \mathbb{R}^{Jk} \to \mathcal{Q} \subset \mathcal{P}^J$, parameterized by $\theta \in \Theta \subseteq \mathbb{R}^p$ :

$$\mathsf{J}(q) = \mathbb{E}^{Y^\dagger}\Big[\mathsf{D}_{\mathrm{KL}}(q\|\rho) - \mathbb{E}^q\big[\log \mathsf{l}(Y^\dagger|V)\big]\Big], \tag{9.14a}$$

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{J}\big(\mathcal{S}_\theta(Y^\dagger)\big). \tag{9.14b}$$

Then, for given unseen data $Y^\dagger$, we choose our approximate smoothing distribution $q^\star = \mathcal{S}_{\theta^\star}(Y^\dagger)$. Consider Subsection 9.3.1 but now assume that the mean of the Gaussian is computed (approximately) as function $V^\star(Y^\dagger)$ mapping observations to the 4DVar solution, as in Section 9.2; and in the context of uncertainty quantification now allow the covariance $\Sigma$ to be a function of $Y^\dagger$ parameterized by $\theta \in \Theta$. Thus $\Sigma = \Sigma(Y^\dagger; \theta)$. See Example 6.3.

## 9.4 Filtering: State Estimation

Recall the dynamics/data model (7.1) and (7.2), repeated here for convenience:

$$v^\dagger_{j+1} = \Psi(v^\dagger_j) + \xi^\dagger_j,\ j \in \mathbb{Z}^+, \tag{9.15a}$$

$$y^\dagger_{j+1} = h(v^\dagger_{j+1}) + \eta^\dagger_{j+1},\ j \in \mathbb{Z}^+. \tag{9.15b}$$

Our interest in this section is in learning filtering algorithms for this problem; we make the following data assumption:

**Data Assumption 9.7.** *We have access to data set* $\{v^\dagger_{j+1}, y^\dagger_{j+1}\}_{j\in\{0,\dots,J-1\}}$ *generated as a realization of* (9.15)*.*

We will study learning problems found as generalization of three specific algorithms from Chapter 7, namely 3DVar, the EnKF and the optimal particle filter. We use the class of distance-like deterministic scoring rules from Definition 12.56. Using a choice of such a rule $\mathsf{D}(\cdot,\cdot)$ we define an objective function from which to find parameters $\theta$ specifying a filtering algorithm.

**Example 9.8.** The canonical example of a distance-like deterministic scoring rule is $\mathsf{D}(v, w) = |v - w|^2$, the squared Euclidean distance. ♢

### 9.4.1 3DVar

**General $\Psi$ and $h$**

Now recall the 3DVar algorithm (7.17), also repeated here for convenience:

$$\widehat{v}_{j+1} = \Psi(v_j), \tag{9.16a}$$

$$v_{j+1} = \widehat{v}_{j+1} + K\big(y^\dagger_{j+1} - h(\widehat{v}_{j+1})\big). \tag{9.16b}$$

For the dynamics/data model and for 3DVar, we make the same Gaussian and independence assumptions on the initialization and noise as made in equations (7.1), (7.2) and Assumption 7.3. In particular we assume that $v_0^\dagger$ and $v_0$ are drawn from the same distribution $\mathcal{N}(m_0, C_0)$, but independently. To fully specify the 3DVar algorithm we need to choose matrix $K$. We investigate learning $\theta := K$ on the basis of data. (We note that, alternatively, $K$ could be parameterized as $K = K(\theta)$, and parameter $\theta$ could be learned.) We emphasize the dependence of the output of 3DVar on unknown parameter $\theta$ by writing $v_j(\theta)$. We apply a scoring rule $\mathsf{D}(\cdot,\cdot)$ satisfying Definition 12.56 to measure the distance between $v_j^\dagger$ and $v_j$, and estimate a $\theta^\star$ that minimizes the score averaged over the time series:

$$\mathsf{J}^J(\theta) = \frac{1}{J}\sum_{j=1}^{J} \mathsf{D}(v_j(\theta), v_j^\dagger), \tag{9.17a}$$

$$\theta^\star \in \arg\min_\theta \mathsf{J}^J(\theta). \tag{9.17b}$$

Note that $v_j(\theta)$ depends on the observed data $\{y_i^\dagger\}_{i\in\{1,\dots,j\}}$ and so we are indeed using the entirety of the data specified in Data Assumption 9.7. The minimization (9.17) may be performed using auto-differentiation with respect to $\theta$; see Section 16.2 for details on this methodology.

Remark 9.9. In defining the previous learning frameworks we have started from the population loss, defined as expectation over a measure with Lebesgue density, and then noted that in practice we approximate this empirically. Here, because the population loss is more complicated to write down, we work the other way around: we have started with empirical loss (9.17) and now proceed to derive a population loss. To this end let us view (9.15), (9.16) as a coupled stochastic dynamical system for $(v_j^\dagger, y_j^\dagger, v_j)$ and assume it is ergodic with invariant measure $\gamma(dv^\dagger, dy^\dagger, dv; \theta)$; note, in particular that this measure depends on $\theta$. Furthermore, note that the measure factorizes naturally as $\gamma(dv^\dagger, dy^\dagger, dv; \theta) = \pi(dv|y^\dagger;\theta)\mathsf{l}(dy^\dagger|v^\dagger)\mu(dv^\dagger)$. We may then view $\mathsf{J}^J(\cdot)$ as approximation of the population-level loss $\mathsf{J}(\cdot)$ found, by ergodicity, in the limit $J \to \infty$ :

$$\mathsf{J}(\theta) = \int \mathsf{D}(v, v^\dagger)\pi(dv|y^\dagger;\theta)\mathsf{l}(dy^\dagger|v^\dagger)\,\mu(dv^\dagger).$$

$\diamondsuit$

### Linear $\Psi$ and $h$

We show that for linear dynamics and observations, taking $\mathsf{D}$ to be the trace of the covariance matrix leads to a gain which converges to the steady-state Kalman gain as $J \to \infty$. Consider the 3DVar algorithm (9.16) with $\Psi(\cdot) = A\cdot$ and $h(\cdot) = H\cdot$. Then $v_j$ obeys the recursion

$$v_{j+1} = (I - KH)\widehat{v}_{j+1} + Ky_{j+1}^\dagger, \tag{9.18a}$$

$$\widehat{v}_{j+1} = Av_j. \tag{9.18b}$$

Define the error covariances $C_j = \mathbb{E}[(v_j^\dagger - v_j)\otimes(v_j^\dagger - v_j)]$ and $\widehat{C}_j = \mathbb{E}[(v_j^\dagger - \widehat{v}_j)\otimes(v_j^\dagger - \widehat{v}_j)]$ which, note, are not the covariances of the filtering distribution. These covariances obey the recursions

$$C_{j+1} = (I-KH)\widehat{C}_j(I-KH)^\top + K\Gamma K^\top, \tag{9.19a}$$
$$\widehat{C}_{j+1} = AC_jA^\top + \Sigma. \tag{9.19b}$$

The recursion for $\widehat{C}_j$ holds because

$$\begin{aligned}\widehat{C}_{j+1} &= \mathbb{E}\Big[(v_{j+1}^\dagger - \widehat{v}_{j+1})\otimes(v_{j+1}^\dagger - \widehat{v}_{j+1})\Big]\\ &= \mathbb{E}\Big[(Av_j^\dagger + \xi_j - Av_j)\otimes(Av_j^\dagger + \xi_j - Av_j)\Big]\\ &= \mathbb{E}\Big[\big(A(v_j^\dagger - v_j)\big)\otimes\big(A(v_j^\dagger - v_j)\big)\Big] + \Sigma\\ &= AC_jA^\top + \Sigma.\end{aligned}$$

The recursion for $C_j$ holds because

$$\begin{aligned}C_{j+1} &= \mathbb{E}\Big[(v_{j+1}^\dagger - v_{j+1})\otimes(v_{j+1}^\dagger - v_{j+1})\Big]\\ &= \mathbb{E}\Big[(v_{j+1}^\dagger - (I-KH)\widehat{v}_{j+1} - KHv_{j+1}^\dagger - K\eta_{j+1})\\ &\qquad \otimes (v_{j+1}^\dagger - (I-KH)\widehat{v}_{j+1} - KHv_{j+1}^\dagger - K\eta_{j+1})\Big]\\ &= \mathbb{E}\Big[(I-KH)(v_{j+1}^\dagger - \widehat{v}_{j+1})\otimes(I-KH)(v_{j+1}^\dagger - \widehat{v}_{j+1})\Big] + K\Gamma K^\top\\ &= (I-KH)\widehat{C}_j(I-KH)^\top + K\Gamma K^\top.\end{aligned}$$

We consider learning $K$ by minimizing

$$\mathsf{J}^J(K) = \lim_{J\to\infty}\frac{1}{J+1}\sum_{j=0}^{J}\mathrm{Tr}(C_j). \tag{9.20}$$

In the following theorem, we show that the minimizer of $\mathsf{J}^J(K)$ converges to the steady-state Kalman gain as $J\to\infty$.

**Theorem 9.10.** *Take the 3DVar algorithm (9.16) with $\Psi(\cdot) = A\cdot$ and $h(\cdot) = H\cdot$. Suppose that the analysis covariance $C_j$ of the Kalman filter converge as $j\to\infty$ to the steady state in (7.15). Then, $K^\star \in \arg\min_K \mathsf{J}^J(K)$, where $\mathsf{J}^J(K)$ is given by (9.20), converges to the steady-state Kalman gain $K_\infty$ (7.14b) as $J\to\infty$.*

*Proof.* We begin by noting that (9.18) and (9.19) also hold for a time-varying gain $K_j$:

$$v_{j+1} = (I-K_jH)Av_j + K_jy_{j+1}^\dagger, \tag{9.21a}$$
$$C_{j+1} = (I-K_jH)\widehat{C}_j(I-K_jH)^\top + K_j\Gamma K_j^\top, \tag{9.21b}$$
$$\widehat{C}_{j+1} = AC_jA^\top + \Sigma. \tag{9.21c}$$

It is readily shown that the $K_j$ that minimizes $C_{j+1}$ given $\widehat{C}_j$ (with respect to the order of positive definite matrices) is the Kalman gain $K_j^\star = \widehat{C}_j H^\top \big(H\widehat{C}_j H^\top + \Gamma\big)^{-1}$. This can be seen by setting $K_j' = K_j^\star + \Delta K$, where $\Delta K$ is an arbitrary matrix, and showing that $(I - K_j' H)\widehat{C}_j(I - K_j' H)^\top + K_j'\Gamma K_j'^\top \geq (I - K_j^\star H)\widehat{C}_j(I - K_j^\star H)^\top + K_j^\star \Gamma K_j^{\star\top}$.

We now introduce the notation $C_j^\star$ to indicate the matrix given by the recursion (9.21) when the Kalman gain is used. We write $C_j(K)$ for the solution to the recursion when the gain is fixed at some $K_j = K$ for all $j$.

It can be seen by induction that $C_j(K) - C_j^\star \geq 0$ for all $j$. Since $\mathrm{Tr}(\cdot)$ is monotonically increasing with respect to the order of positive definite matrices, we have that

$$\mathrm{Tr}(C_j^\star) \leq \mathrm{Tr}\big(C_j(K)\big)$$

for any fixed $K$ and all $j$.

Taking the limit $j \to \infty$, we have that for all $K$

$$\lim_{j\to\infty} \mathrm{Tr}(C_j^\star) = \mathrm{Tr}\big(C_\infty(K_\infty)\big) \leq \lim_{j\to\infty} \mathrm{Tr}\big(C_j(K)\big),$$

where we have used the convergence of the Kalman filter to its steady state. Thus, $\mathrm{Tr}\big(C_j(K)\big)$ is minimized at $K = K_\infty$, the steady-state Kalman gain.

By the Stolz–Cesàro theorem (see bibliography), we have that

$$\lim_{J\to\infty} \frac{1}{J+1}\sum_{j=0}^{J} \mathrm{Tr}(C_j^\star) = \lim_{J\to\infty} \frac{1}{J+1}\sum_{j=0}^{J} \mathrm{Tr}\big(C_\infty(K_\infty)\big) \leq \lim_{J\to\infty} \frac{1}{J+1}\sum_{j=0}^{J} \mathrm{Tr}\big(C_j(K)\big)$$

for all $K$. □

### 9.4.2 EnKF Gain

Although the ensemble Kalman method was designed as a Monte Carlo method, it is often used as a state estimator and the ensemble is used to estimate uncertainty in the state estimates. This is the perspective we adopt here. We continue to work under Data Assumption 9.7 and recall the EnKF algorithm (7.23). Rather than calculating $K_{j+1}$ from empirical covariances as in (16.15), we instead try and learn dependence on the ensemble. To this end we modify (7.23) to read

$$\widehat{v}_{j+1}^{(n)} = \Psi(v_j^{(n)}) + \xi_j^{(n)}, \quad n = 1, \dots, N, \tag{9.22a}$$

$$v_{j+1}^{(n)} = \widehat{v}_{j+1}^{(n)} + K_{j+1}\big(y_{j+1}^\dagger - \eta_{j+1}^{(n)} - h(\widehat{v}_{j+1}^{(n)})\big), \quad n = 1, \dots, N, \tag{9.22b}$$

$$K_{j+1} = \mathsf{K}(\widehat{v}_{j+1}^{(1)}, \dots, \widehat{v}_{j+1}^{(N)}; \theta), \tag{9.22c}$$

where $\xi_j^{(n)} \sim \mathcal{N}(0, \Sigma)$, $\quad \eta_{j+1}^{(n)} \sim \mathcal{N}(0, \Gamma)$ are independent sequences of i.i.d. random vectors with respect to both $j$ and $n$, and the two sequences themselves are independent of one another. Here, in this setting, it may be desirable that $\mathsf{K}(\cdot;\theta)$ be invariant with respect to permutation of the ensemble.

The assumptions on the noise are as detailed after (7.23). It is useful to define a state estimator by taking the mean of the ensemble at time $j$:

$$\overline{v}_j = \frac{1}{N}\sum_{n=1}^{N} v_j^{(n)}. \tag{9.23}$$

Remark 9.11. It may be of interest to replace (9.22c) by $K_{j+1} = \mathsf{K}(\widehat{C}^{vh}_{j+1}, \widehat{C}^{yy}_{j+1}; \theta)$, where the covariance matrices are as defined in (16.15) and $\mathsf{K}(\cdot;\theta)$ is a parameterized family of gain functions to be learned by optimizing over $\theta$. For example $\mathsf{K}(\cdot;\theta)$ may be parameterized as a neural network. Recall from Remark 7.11 that if $h$ is linear then the two covariances $\widehat{C}^{vh}_{j+1}, \widehat{C}^{yy}_{j+1}$ can be expressed in terms of $\widehat{C}^{vv}_{j+1}$ and $H$; thus in this case we might seek to learn the gain in the form $\mathsf{K}(\widehat{C}^{vv}_{j+1}, H; \theta)$. Furthermore it may also be of interest to replace (9.22b,9.22c) by an update of the form

$$\begin{aligned}
v_{j+1}^{(n)} &= \mathsf{k}(\widehat{v}_{j+1}^{(n)}, \overline{w}_{j+1}, i_{j+1}^{(n)}, \widehat{C}^{vh}_{j+1}, \widehat{C}^{yy}_{j+1}, \widehat{C}^{vv}_{j+1}; \theta), \quad n = 1, \dots, N,\\
i_{j+1}^{(n)} &= y_{j+1}^{\dagger} - \eta_{j+1}^{(n)} - h(\widehat{v}_{j+1}^{(n)}),
\end{aligned}$$

where

$$\overline{w}_{j+1} = \frac{1}{N}\sum_{n=1}^{N} \widehat{v}_{j+1}^{(n)}$$

is the mean of the predicted ensemble. The reader will be able to suggest many variants on the preceding formulations of a learning problem for an EnKF-like data assimilation algorithm. ♢

Again, to emphasize dependence of the algorithm on parameters $\theta$ to be learned, we write $v_j^{(n)}(\theta)$ and the state estimator $\overline{v}_j(\theta)$. We then define $\theta$ by

$$\mathsf{J}^J(\theta) = \frac{1}{J}\sum_{j=1}^{J} \mathsf{D}(\overline{v}_j(\theta), v_j^{\dagger}). \tag{9.24a}$$

$$\theta^\star \in \arg\min_\theta \mathsf{J}^J(\theta). \tag{9.24b}$$

Here $\mathsf{D}(\cdot,\cdot)$ is again a deterministic scoring rule from Definition 12.56.

Remark 9.12. Instead of minimizing the error of the ensemble mean $\overline{v}_j$ with respect to the true state $v_j^{\dagger}$ using a deterministic scoring rule, one can also desire that the distribution of the ensemble be informative of the error with respect to the truth. For example, it is possible to ask that the ensemble spread match the expected error with respect to the truth (see Subsection 9.4.5 for more discussion of this point). Considering the ensemble as an empirical measure, $\pi_{j+1} = \frac{1}{N}\sum_{n=1}^{N}\delta_{v_{j+1}^{(n)}}$, instead of the cost function (9.24) one based on a probabilistic scoring rule (see Section 12.3) can be applied:

$$\mathsf{J}^J(\theta) = \frac{1}{J}\sum_{j=1}^{J} \mathsf{D}(\pi_j(\theta), v_j^{\dagger}).$$

♢

### 9.4.3 EnKF Localization and Inflation

Inflation and localization, discussed in Subsections 7.6.2 and 7.6.3, are essential for the performance of the EnKF, as discussed in Remark 7.12. Learning can be used to determine appropriate parameters for these features of the EnKF. Again we continue to work under Data Assumption 9.7.

Recall multiplicative inflation given by (7.28),

$$\widehat{v}_{j+1}^{(n)} \to \widehat{m}_{j+1} + \alpha(\widehat{v}_{j+1}^{(n)} - \widehat{m}_{j+1}),$$

and covariance localization given by (7.29), (7.30),

$$\begin{aligned}\widehat{C}_{j+1} &\to L \circ \widehat{C}_{j+1},\\ (L)_{ik} &= e^{-D_{ik}^2/\ell}.\end{aligned}$$

The inflation parameter $\alpha$ and localization radius $\ell$ can be considered parameters $\theta = (\alpha, \ell)$, and then optimized as in Subsection 9.4.2.

Remark 9.13. Other ways of parameterizing localization could also be considered, such as taking $\theta = L$ and learning the entire matrix, or parameterizing $L(\theta)$ with a given structure different from the one given above. Alternatives to Schur product localization can also be considered, such as ones that learn a nonlinear map that localizes the gain,

$$K_j \to \mathsf{L}(K_j; \theta).$$

♢

### 9.4.4 Optimal Particle Filter

Here we work in the setting of linear observation operator, (7.33), repeated here for convenience:

$$\begin{aligned}v_{j=1}^{\dagger} &= \Psi(v_j^{\dagger}) + \xi_j^{\dagger},\\ y_{j+1}^{\dagger} &= Hv_{j+1}^{\dagger} + \eta_{j+1}^{\dagger}\,.\end{aligned}$$

We again make the same Gaussian and independence assumptions on the initialization and noise as are made in equations (7.1), (7.2) and Assumption 7.3. Recall that the optimal particle filter, from Section 7.8, works by proposing particles through an ensemble of equally weighted noisy 3DVars and then reweighting them: see equation (7.38). Here, recognizing that reweighting often leads to particle collapse (see Section 7.12), we instead seek to learn a desirable combination of the ensemble of 3DVars:

$$\widehat{v}_{j+1}^{(n)} = (I - KH)\Psi(v_j^{(n)}) + Ky_{j+1}^{\dagger} + \xi_{n+1}^{(n)}, \tag{9.25a}$$

$$v_{j+1}^{(n)} = \widehat{v}_{j+1}^{(n)} + \mathsf{k}\Big(\{\widehat{v}_{j+1}^{(n)}\}_{n=1}^{N}, \widehat{v}_{j+1}^{(n)}, y_{j+1}^{\dagger}, i_{j+1}^{(n)}; \theta\Big), \quad n = 1, \dots, N, \tag{9.25b}$$

$$i_{j+1}^{(n)} = y_{j+1}^{\dagger} - H\widehat{v}_{j+1}^{(n)}. \tag{9.25c}$$

with $\xi_{j+1}^{(n)}$ i.i.d. in $j$ and $n$ and distributed according to $\mathcal{N}(0, C)$ and matrix $K$ as defined in (7.36). Again it may be desirable that $\mathsf{k}(\cdot, \widehat{v}_{j+1}^{(n)}, y_{j+1}^{\dagger}, i_{j+1}^{(n)}; \theta)$ is invariant with respect to permutation of the ensemble.

We continue to work under Data Assumption 9.7. To emphasize dependence of the algorithm on parameters $\theta$ to be learned, we again write $v_j^{(n)}(\theta)$ and the state estimator from (9.23) as $\overline{v}_j(\theta)$. We then determine $\theta$ by (9.24). As in the context of the EnKF, other forms of learning problem may be postulated; see Remark 9.11.

### 9.4.5 Spread–Error Relationship

In this subsection we discuss the connection between scoring rules and the *spread–error ratio*, another distance-like function used to evaluate forecasts. Note that we have, in this section, trained methods as state estimators, not to give probabilistic forecasts. Nonetheless it is possible to ask, for the particle-based methods, what properties one might desire from the whole ensemble as well as from its mean. One answer to this question is to consider the spread–error ratio. To this end we view ensemble members $\{v_j^{(n)}\}_{n=1}^N$ as providing an approximate filtering distribution

$$\pi_j(\theta) = \frac{1}{N} \sum_{n=1}^{N} \delta_{v_j^{(n)}}.$$

To motivate the following definition, read the statement of Theorem 7.6 which explains a key property of filtering distributions which is a straightforward consequence of conditional expectation.

**Definition 9.14.** Consider a sequence of probabilistic forecasts $\{\pi_j\}_{j=1}^J$. The *spread–error ratio* of this sequence with respect to the true trajectory $\{v_j^{\dagger}\}_{j=1}^J$ is the ratio

$$r = \frac{\frac{1}{J}\sum_{j=1}^{J} \mathbb{E}^{v_j \sim \pi_j}\left[|v_j - \mathbb{E}^{v_j \sim \pi_j}[v_j]|^2\right]}{\frac{1}{J}\sum_{j=1}^{J} \left|v_j^{\dagger} - \mathbb{E}^{v_j \sim \pi_j}[v_j]\right|^2}.$$

If $r = 1$, we say that the forecasts $\{\pi_j\}_{j=1}^J$ have a *perfect spread–error relationship*. If $r < 1$, we say that they are *underdispersed*. If $r > 1$, we say that they are *overdispersed*.[1] $\diamondsuit$

For a given trajectory error, does minimizing a probabilistic scoring rule with respect to covariance recover a spread–error ratio of 1? We show that it does with the Dawid–Sebastiani score and the logarithmic scoring rule with a Gaussian forecast, but not in the case of the continuous ranked probability score (CRPS).

**Theorem 9.15.** *For $1 \le j \le J$, let $m_j = \mathbb{E}^{v_j \sim \pi_j}[v_j]$ and let*

$$e = \frac{1}{J} \sum_{j=1}^{J} (m_j - v_j^{\dagger})^2.$$

[1] Note that a different ratio other than 1 can be used. The important point is that the spread should vary in proportion to the error.

*Then, in the scalar case, the variances $C_j = \mathbb{E}^{v_j \sim \pi_j}\big[(v_j - m_j)^2\big]$ that minimize the sum of the Dawid–Sebastiani scores, $\{C_j^\star\}_{j=1}^J$, are such that $\frac{1}{J}\sum_{j=1}^J C_j^\star = e$, thus having a spread–error ratio of 1.*

*Proof.* The sum of the Dawid–Sebastiani scores (12.34) over the trajectory is given in the scalar case by

$$\sum_{j=1}^{J}\left[\frac{(v_j^\dagger - m_j)^2}{C_j} + \log(C_j)\right].$$

Minimizing each term with respect to $C_j$, we find that $C_j^\star = (v_j^\dagger - m_j)^2$. Thus, $\frac{1}{J}\sum_{j=1}^J C_j^\star = e$. □

Remark 9.16. From the above proof, it is straightforward to see that this theorem is also true when the Dawid–Sebastiani score is replaced by the logarithmic scoring rule (12.32), but restricted to Gaussian $\pi_j$. ♢

**Proposition 9.17.** *Theorem 9.15 is not true when the Dawid–Sebastiani rule is replaced with the CRPS* (12.22)*, and thus, also the energy score. Furthermore, the spread–error ratio of the optimal forecast is not constant when $|\mathbb{E}[v] - v^\dagger|^2$ is varied.*

*Proof.* Proof by example. Set $J = 1$, $v^\dagger = 1$, and $m = 0$, so that $e = 1$. Then it can be found numerically that $\mathsf{CRPS}(\pi, v^\dagger)$ is minimized when $C \approx 1.44$, so that $r \approx 1.44$. When $v^\dagger = 2$ and $m = 0$, $e = 4$, and the CRPS is minimized at $C \approx 1.50$, so that $r \approx 0.38$. □

### 9.4.6 Learning State Estimators in the Presence of Model Error

We consider the setting in which the model (9.15) is imperfect and hence we seek to learn state estimation data assimilation algorithms only from data; no access to the true state is assumed. We thus employ:

**Data Assumption 9.18.** *We have access to data set $\{y_{j+1}^\dagger\}_{j\in\{0,\dots,J-1\}}$ generated from a realization of* (9.15)*.*

We then use deterministic scoring rules from Definition 12.56, but applied in the observation space rather than in the state space. Learning data assimilation algorithms, as in the previous section, but based on an imperfect model trajectory, may not be optimal. Instead we assume that the observation model (7.2) is perfect, but for a true data signal which does not come from (7.1); thus we have access only to $\{y_{j+1}^\dagger\}_{j\in\mathbb{Z}^+}$. We do assume, however, that we know the function $h(\cdot)$ mapping from the unknown true state to the data space. In this setting, focusing on 3DVar, we can consider the loss

$$\mathsf{J}^J(\theta) = \frac{1}{J}\sum_{j=1}^{J}\mathsf{D}\Big(h(v_j(\theta), y_j^\dagger\Big) \tag{9.26}$$

instead of (9.17). Note that $\mathsf{J}^J(\theta)$ is then a proxy for the loss with respect to the true trajectory in the observation space:

$$\frac{1}{J}\sum_{j=1}^{J}\mathsf{D}\Big(h(v_j(\theta)), h(v_j^\dagger)\Big). \tag{9.27}$$

These ideas also enable us generalize to the loss function (9.24) to the setting of EnKF and the optimal particle filter.

**Example 9.19.** The cost function $\mathsf{J}^J(\theta)$ in (9.26) may be problematic; since $v_j$ is a function of $y_j^\dagger$, it can be overfit to observations. For example, if $h$ is surjective, one can achieve a perfect score by simply setting $v_j = h^{-1}(y_j^\dagger)$, where $h^{-1}$ is the right inverse of $h$. We describe a method to avoid overfitting when using the squared Euclidean distance as the scoring rule. Similar analyses could be done for other scoring rules.

Take $\mathsf{D}(\cdot,\cdot)$ to be the squared Euclidean norm: $\mathsf{D}(v, w) = |v - w|^2$. We analyze a single time $j$. Taking the expectation over the observation noise realization, $\eta_j^\dagger \sim \mathcal{N}(0, \Gamma)$, we obtain:

$$\mathbb{E}\Big[|y_j^\dagger - h(v_j(\theta))|^2\Big] = \mathbb{E}\Big[|h(v_j^\dagger) - h(v_j)|^2\Big] + \mathrm{Tr}(\Gamma) - 2\mathbb{E}\Big[h\big(v_j(\theta)\big)^\top \eta_j^\dagger\Big].$$

Note that $\mathbb{E}\Big[h(v_j)^\top \eta_j^\dagger\Big]$ will generally be positive, since the observation $y_j^\dagger$ was used to produce the analysis $v_j$. The first term on the RHS is (9.27), which is what we would like to estimate. Thus, $\mathbb{E}\Big[|y_j^\dagger - h(v_j(\theta))|^2\Big]$ will systematically underestimate $\mathbb{E}\Big[|h(v_j^\dagger) - h(v_j)|^2\Big] + \mathrm{Tr}(\Gamma)$. The term $2\mathbb{E}\Big[h\big(v_j(\theta)\big)^\top \eta_j^\dagger\Big]$ is called the *optimism.*

$\mathbb{E}\Big[|y_j^\dagger - h(v_j(\theta))|^2\Big] + 2\mathbb{E}[h\big(v_j(\theta)\big)^\top \eta_j^\dagger]$ is thus a better proxy for out-of-sample performance, and should be minimized instead of only $\mathbb{E}\Big[|y_j^\dagger - h(v_j(\theta))|^2\Big]$. ♢

## 9.5 Filtering: Probabilistic Estimation

In this section we discuss how classes of algorithms may be optimized to approximate probabilistic estimation defined by the filtering distribution. We concentrate on particle-based methods.

Recall that the time evolution of the filtering distribution can be defined by interweaving prediction by the underling stochastic dynamics with Bayes Theorem to incorporate the observations via the analysis step – see Section 7.1. In particular (7.4) defines the evolution via the (i) prediction $\pi_j \mapsto \widehat{\pi}_{j+1}$ and the (ii) analysis $\widehat{\pi}_{j+1} \mapsto \pi_{j+1}$ steps; it is in step (ii) that the data $y_{j+1}^\dagger$ is incorporated. The bootstrap particle filter of Section 7.7 uses this factorization of the filter evolution. The optimal particle filter of Section 7.8 uses a different factorization into (i) $\pi_j \mapsto \mathfrak{p}_{j+1}$ and (ii) $\mathfrak{p}_{j+1} \mapsto \pi_{j+1}$, in which both steps depends on the data $y_{j+1}^\dagger$.

For both particle filters an issue arising in application to high dimensional problems is weight collapse. We provide a methodology to learn new ensemble methods that lead to equal weight ensemble filters, trained to be close to the bootstrap or optimal particle filters in step (ii). Our available data is as follows:

**Data Assumption 9.20.** *We are given i.i.d. samples $\{\widehat{v}_{j+1}^{(n)}\}_{n=1}^N$ from the forecast distribution $\widehat{\pi}_{j+1}$ at time $j+1$.*

In what follows we use distance measures, such as those derived from the scoring rules from Section 12.3. (We exclude the deterministic scoring rules from Subsection 12.3.7, given in Definition 12.56, that are employed solely in the context of state estimation.) The key attribute we seek for the distance is that it is implementable given only samples. Indeed it is instructive to think of scoring rules as being introduced for this purpose: metrics and divergences may not be amenable to measuring distance between two probability distributions when both are given only through samples. In Subsection 9.5.1 we discuss who scoring rules may be used to learn probabilitistic filters. Subsection 9.5.2 then deploys these ideas in the context of learning using the bootstrap particle filter; Subsection 9.5.3 generalizes to the optimal particle filter.

### 9.5.1 Learning Filters Using Strictly Proper Scoring Rules

Recall the dynamical system (7.1)

$$\begin{aligned} v_{j+1}^\dagger &= \Psi(v_j^\dagger) + \xi_j^\dagger, \\ v_0^\dagger &\sim \mathcal{N}(m_0, C_0), \quad \xi_j^\dagger \sim \mathcal{N}(0, \Sigma) \text{ i.i.d.}, \end{aligned}$$

with observations given by (7.2)

$$\begin{aligned} y_{j+1}^\dagger &= h(v_{j+1}^\dagger) + \eta_{j+1}^\dagger, \\ \eta_j^\dagger &\sim \mathcal{N}(0, \Gamma) \quad \text{i.i.d.} \end{aligned}$$

Also recall the notation, for a given and fixed integer $J$,

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

**Theorem 9.21.** *Consider a strictly proper scoring rule $S(\cdot,\cdot)$. Define $\mathsf{J}_j : \mathcal{P} \to \mathbb{R}$ by*

$$\mathsf{J}_j(q) = \mathbb{E}^{v_j^\dagger \sim \mathbb{P}(v_j^\dagger | Y_j^\dagger)} \Big[ S(q, v_j^\dagger) \Big], \tag{9.28a}$$

$$q_{\text{OPT}} \in \arg\min_{q \in \mathcal{P}} \mathsf{J}(q). \tag{9.28b}$$

*Then $q_{\text{OPT}} = \mathbb{P}(v_j^\dagger | Y_j^\dagger)$, the joint distribution defined by the filtering distribution given in (7.1), (7.2).*

*Proof.* This is a direct consequence of the definition of a strictly proper scoring rule – see Definition 12.35. □

Thus, having only the information $Y_j^\dagger$, the filtering distribution minimizes the expected score. Hence $\mathsf{J}_j(\cdot)$ may be used as the basis for learning algorithms designed to match the true filter, using only samples from the proposed approximate filter. Certain integral probability metrics are amenable to sample-based implementation, which will be convenient in what follows.

**Corollary 9.22.** *Consider a strictly proper scoring rule $S(\cdot,\cdot)$. Define $\mathsf{J}:\mathcal{P}\to\mathbb{R}$ by, for $q=(q_1,\cdots,q_J)$,*

$$\mathsf{J}(q)=\frac{1}{J}\sum_{j=1}^{J}\mathbb{E}^{(v_j^\dagger,Y_j^\dagger)}\Big[S(q_j,v_j^\dagger)\Big], \tag{9.29a}$$

$$q_{\mathrm{OPT}}\in\arg\min_{q\in\mathcal{P}^J}\mathsf{J}(q). \tag{9.29b}$$

*Then $q_{\mathrm{OPT}}=(q_{\mathrm{OPT}1},\cdots,q_{\mathrm{OPT}J})$ where $q_{\mathrm{OPT}j}=\mathbb{P}(v_j^\dagger|Y_j^\dagger)$, the filtering distribution defined by (7.1), (7.2).*

*Proof.* This follows from the preceding theorem by averaging over the marginal on $Y_j^\dagger$ defined by the filtering distribution defined by (7.1), (7.2), and then averaging over time index $j$. □

We now parameterize the set of candidate approximate filters $q$ to develop the basis of actionable algorithms. Recall the prediction and analysis operators defined in (7.4). Consider a family of algorithms in which the analysis step is parameterized by $\theta$, yielding the following recursion:

$$q_{j+1}(\theta)=\mathsf{A}_\theta(\mathsf{P}q_j(\theta);y_{j+1}^\dagger),\quad q_0=\Pi_0. \tag{9.30}$$

**Corollary 9.23.** *Consider $q_j$ evolving under $\theta$-parameterized algorithm (9.30) leading to algorithm $q(\theta)\in\mathcal{P}^J$. Then, we can write (9.29a) as $\mathsf{I}(\theta)=\mathsf{J}\big(q(\theta)\big)$ to obtain*

$$\mathsf{I}(\theta)=\frac{1}{J}\sum_{j=1}^{J}\mathbb{E}^{(v_j^\dagger,y_j^\dagger)}\Big[S(\mathsf{A}_\theta(\mathsf{P}q_{j-1}(\theta);y_j^\dagger),v_j^\dagger)\Big]. \tag{9.31}$$

Remark 9.24. Because the update in (9.30) depends only on data $y_j^\dagger$ the expectation defining $\mathsf{I}(\cdot)$ is over $(v_j^\dagger,y_j^\dagger)$ whereas for the general $\mathsf{J}(\cdot)$ it is over $(v_j^\dagger,Y_j^\dagger)$. This makes implementation of empirical minimization algorithms, based on $\mathsf{I}(\theta)$, straightforward in practice. For example it is possible to replace the expectation $\mathbb{E}^{(v_j^\dagger,y_j^\dagger)}$ by a single realization and then take $J$ large and appeal to ergodicity. ♢

### 9.5.2 Bootstrap Particle Filter

Recall, from (7.31), (7.32), the particle filter approximation of the filtering distribution $\pi_{j+1}$:

$$\pi_{j+1}^{\mathrm{PF}}=\sum_{m=1}^{M}w_{j+1}^{(m)}\delta_{\widehat{v}_{j+1}^{(m)}}.$$

Here the particles $\widehat{v}_j^{(m)}$ and weights $w_j^{(m)}$ evolve according to

$$\begin{aligned}\widehat{v}_{j+1}^{(m)}&=\Psi(v_j^{(m)})+\xi_j^{(m)},\\ \ell_{j+1}^{(m)}&=\exp\Big(-\frac{1}{2}|y_{j+1}^\dagger-h(\widehat{v}_{j+1}^{(m)})|_\Gamma^2\Big),\\ w_{j+1}^{(m)}&=\ell_{j+1}^{(m)}\Big/\Big(\sum_{i=1}^{M}\ell_{j+1}^{(i)}\Big).\end{aligned}$$

We have chosen integer $M$ here for the number of particles to emphasize that it may be different from integer $N$ used in the ensuing ensemble Kalman-like method. From $\pi_{j+1}^{\mathrm{PF}}$ we may construct an equally weighted approximation by resampling to obtain

$$\pi_{j+1}^{\mathrm{RPF}} = \sum_{m=1}^{M} \frac{1}{M} \delta_{v_{j+1}^{(m)}}.$$

Here the $v_{j+1}^{(m)}$ are drawn i.i.d. from $\pi_{j+1}^{\mathrm{PF}}$. In this process some of the $\{\widehat{v}_{j+1}^{(r)}\}_{r=1}^{M}$ may be dropped and others repeated.

We want to learn parameters $\theta$ in a modified ensemble Kalman filter, defined by equation (9.22), so that the following equally weighted approximation of the filtering distribution is close to the true filtering distribution:

$$\pi_{j+1}^{\mathrm{EnKF}}(\theta) = \sum_{n=1}^{N} \frac{1}{N} \delta_{v_{j+1}^{(n)}(\theta)}.$$

Here, recalling (9.22),

$$\begin{aligned}
\widehat{v}_{j+1}^{(n)} &= \Psi(v_j^{(n)}) + \xi_j^{(n)}, \quad n = 1, \dots, N, \\
v_{j+1}^{(n)}(\theta) &= \widehat{v}_{j+1}^{(n)} + K_{j+1}(y_{j+1}^{\dagger} - \eta_{j+1}^{(n)} - h(\widehat{v}_{j+1}^{(n)})), \quad n = 1, \dots, N, \\
K_{j+1} &= \mathsf{K}(\widehat{v}_{j+1}^{(1)}, \dots, \widehat{v}_{j+1}^{(N)}; \theta).
\end{aligned}$$

Recall the integral probability metric MMD from Definition 12.14, noting that it may be implemented for empirical (equally weighted ensemble) measures as in (12.9). We then define $\theta^\star$ by

$$\mathsf{J}^{M,N}(\theta) = \mathsf{D}_{\mathrm{MMD}}(\pi_{j+1}^{\mathrm{RPF}}, \pi_{j+1}^{\mathrm{EnKF}}), \tag{9.32a}$$

$$\theta^\star \in \arg\min_{\theta} \mathsf{J}^{M,N}(\theta). \tag{9.32b}$$

**Remark 9.25.** Note that we have been silent about how the particles $\{v_j^{(m)}\}_{m=1}^{M}$ for the bootstrap particle filter, and the particles $\{v_j^{(n)}\}_{n=1}^{N}$ for the equally weighted ensemble filter, are chosen. They should be chosen from the same distribution $\pi_j$, but they do not need to be identical as points. It is possible to choose $\pi_j = \pi_j^{\mathrm{EnKF}}$ and to average $\mathsf{J}(\cdot)$ over $j$. The preceding methodology could then be implemented under Data Assumption 9.18.

Recall also that $M$ may not equal $N$; indeed it may be advantageous to choose $M \gg N$ to ensure that the resampling process provides good representation of the true filter $\pi_{j+1}$. ♢

### 9.5.3 Optimal Particle Filter

We now describe a similar methodology to that in the preceding subsection, but based on the optimal particle filter rather than the bootstrap particle filter. Recall, from (7.37), (7.38), the optimal particle filter approximation of the filtering distribution $\pi_{j+1}$:

$$\pi_j^{\mathrm{OPF}} = \sum_{m=1}^{M} w_j^{(m)} \delta_{\widehat{v}_j^{(m)}},$$

where the particles $\widehat{v}_j^{(m)}$ and weights $w_j^{(m)}$ evolve according to, for $\xi_{n+1}^{(m)}$ i.i.d. $\mathcal{N}(0, C)$,

$$
\begin{aligned}
\widehat{v}_{j+1}^{(m)} &= (I - KH)\Psi(v_j^{(m)}) + K y_{j+1}^{\dagger} + \xi_{j+1}^{(m)}, \qquad v_j^{(m)} \overset{\text{i.i.d.}}{\sim} \pi_j^{\mathrm{OPF}},\\
\ell_{j+1}^{(m)} &= \exp\Bigl(-\frac{1}{2}|y_{j+1}^{\dagger} - H\Psi(v_j^{(m)})|_S^2\Bigr),\\
w_{j+1}^{(m)} &= \ell_{j+1}^{(m)} \Big/ \sum_{r=1}^{N} \ell_{j+1}^{(r)}.
\end{aligned}
$$

From $\pi_{j+1}^{\mathrm{OPF}}$ we may construct an equally weighted approximation by resampling to obtain

$$
\pi_{j+1}^{\mathrm{ROPF}} = \sum_{m=1}^{M} \frac{1}{M} \delta_{v_{j+1}^{(m)}}.
$$

Here the $v_{j+1}^{(m)}$ are drawn i.i.d. from $\pi_{j+1}^{\mathrm{OPF}}$.

Recall Subsection 9.4.4. We want to learn parameters $\theta$ in a modified ensemble 3DVar, defined by equation (9.22), so that the following equally weighted approximation of the filtering distribution is close to the true filtering distribution:

$$
\pi_{j+1}^{\mathrm{EOPF}}(\theta) = \sum_{n=1}^{N} \frac{1}{N} \delta_{v_{j+1}^{(n)}(\theta)}.
$$

Here, recalling (9.25),

$$
\begin{aligned}
\widehat{v}_{j+1}^{(n)} &= (I - KH)\Psi(v_j^{(n)}) + K y_{j+1}^{\dagger} + \xi_{n+1}^{(n)},\\
v_{j+1}^{(n)} &= \widehat{v}_{j+1}^{(n)} + \mathsf{k}\Bigl(\{\widehat{v}_{j+1}^{(n)}\}_{n=1}^{N}, \widehat{v}_{j+1}^{(n)}, y_{j+1}^{\dagger}, i_{j+1}^{(n)}; \theta\Bigr), \quad n = 1, \dots, N,\\
i_{j+1}^{(n)} &= y_{j+1}^{\dagger} - H\widehat{v}_{j+1}^{(n)}.
\end{aligned}
$$

We then define $\theta^\star$ by

$$
\mathsf{J}^{M,N}(\theta) = \mathsf{D}_{\mathrm{MMD}}(\pi_{j+1}^{\mathrm{EOPF}}, \pi_{j+1}^{\mathrm{ROPF}}), \tag{9.33a}
$$

$$
\theta^\star \in \arg\min_{\theta} \mathsf{J}^{M,N}(\theta). \tag{9.33b}
$$

The same comments as in Remark 9.25 apply here too.

## 9.6 Bibliography

Several works have studied variational formulations of smoothing and filtering problems. An alternative variational objective for jointly learning a parameterized filter and the system dynamics is given in [37]. A variational formulation of the filtering and smoothing problems in continuous time, where the posterior is restricted to be in the exponential family, is provided in [303]. The variational formulation of the optimal continuous-time filter (the Kushner–Stratonovich equation) with no dynamics is derived in [186]. The latter is formally combined with a variational formulation of the Fokker–Planck equation for gradient systems in [179], and it is shown that the solution coincides with the Kalman–Bucy filter in the linear case. Variational filtering is discussed in [209], and the proof of Theorem 9.3 is given there.

We add some remarks on implementation issues arising in Section 9.2. In order to evaluate $\mathsf{J}(\theta)$, the expectations can be replaced by empirical means. Additionally, the KL divergence must be computed. In some cases, such as with Gaussian densities, a closed form of the divergence is available: see Example 12.30, and identity (12.14) in particular. Indeed Gaussian approximation is, in general, popular both because the interpretability of the mean and covariance, and because of the closed form for the divergence. In case only samples of $q_\theta^{Y^\dagger}$ are available (such as in an ensemble smoother), the divergence may be approximated by first applying density estimation to these samples. Since the derivative of $\mathsf{J}(\theta)$ will generally be hard to obtain analytically, auto-differentiation can be applied, and then gradient-based methods used to minimize it. An example of using auto-differentiation in a similar context is provided in [194].

3DVar gain learning was considered in [142, 207], and a lower-dimensional parameterization was considered in [143]. It was also considered in [194] in the case of nonlinear dynamics. It was proven in [142, 207] that the fixed gain which minimizes the expected error with respect to observations, in the asymptotic limit, is the steady-state Kalman gain. However, the cost functions considered were different, and [142] required assumptions on the rank of $H$.

The learning of a neural network analysis step by minimizing a state estimation loss in an EnKF was considered in [217]. The analysis step there is EnKF-like in the sense that the ensemble members interact during the analysis step only through the ensemble mean and covariance. The issue of estimating optimism and out-of-sample performance of DA algorithms is discussed in [40, 207]. Various approaches to learning inflation are discussed in [8, 221]. Learning localization by minimizing the squared Euclidean distance between the analysis and the true trajectory is considered in [330], while a similar approach with a different scoring rule is considered in [223]. Other approaches to learning localization are discussed in [252, 59, 170, 255, 322]. Emulating the analysis step of a Kalman filter using recurrent neural networks is considered in [135]. An ensemble estimator for the energy score was introduced in [116]. For the statement and proof of the Stolz–Cesàro theorem, see Theorem 1.23 in [225].

The use of Gaussian approximations in filtering, including via variational Bayes, is studied in the papers [69, 326, 13]. Combinations with ensemble methods are discussed in [327]. The paper [102] employs mean-field models, while [203] considers learning optimal particle filters using variational inference.

For work on learning controllers, a subject closely related to learning data assimilation algorithms in the presence of model error, see [96, 73].

Learning 4DVar was considered in [97]. A data assimilation approach via reinforcement learning can be found in [128].

# Chapter 10

## Learning the Filter or Smoother Using Transport

The focus of this chapter is the development of measure transport approaches for the probabilistic solution of filtering and smoothing problems. These approaches extend the parameterized maps considered in Chapter 9 for probabilistic estimation, which maintained the structure of classic ensemble filtering and smoothing algorithms. In this chapter, we will find transformations that are not constrained to specific functional forms and can be used to solve general filtering and smoothing problems. Sections 10.1 and 10.2 are analogous to Chapters 5 and 6, respectively, for the transport approach to solving inverse problems. Here, we apply these ideas in the context of the analysis component of filtering. In Section 10.1, we discuss how to build maps from the forecast to an analysis distribution using deterministic transports, while Section 10.2 seeks transports by learning their dependence on the observation. Section 10.3 extends the transport framework to the solution of smoothing problems. The bibliography in Section 10.4 concludes with other instances of transport-based ensemble filtering and smoothing algorithms.

## 10.1 Learning the Forecast to Analysis Map

In this section we seek a transport map that pushes forward the prediction to the analysis distribution in each step of a filtering problem. Following the notation in Chapter 9, we let $\widehat{v}_{j+1} \in \mathbb{R}^d$ denote the state at time $j+1$ following the forecast distribution $\widehat{\pi}_{j+1}$. We would like to find samples from the analysis distribution $\pi_{j+1}$. Our goal is to find a transport $T(\cdot; y^\dagger_{j+1}, \theta)\colon \mathbb{R}^d \to \mathbb{R}^d$, that depends on the observation $y^\dagger_{j+1} \in \mathbb{R}^k$ and parameters $\theta \in \Theta \subseteq \mathbb{R}^p$, and has the property that

$$T(\widehat{v}_{j+1}; y^\dagger_{j+1}, \theta) \sim \pi_{j+1}, \qquad \widehat{v}_{j+1} \sim \widehat{\pi}_{j+1}.$$

After finding $T$, we can evaluate the map at forecast samples to generate analysis samples. Our goal is to find this map given only a forecast ensemble and the observation model. Because it is notationally convenient to do so, we will sometimes drop the dependence of $T$ on $y^\dagger_{j+1}$ in what follows within this section.

**Data Assumption 10.1.** *We are given i.i.d. samples $\{\widehat{v}^{(n)}_{j+1}\}_{n=1}^N$ from the forecast distribution $\widehat{\pi}_{j+1}$ at time $j+1$ and we are able to evaluate $\mathsf{l}(y^\dagger_{j+1}|\widehat{v}^{(n)}_{j+1})$.*

Remark 10.2. In practice, when using methods based on $T$ within multiple iterations of the predict–analysis cycle, we will not have access to *exact* samples from the forecast distribution $\widehat{\pi}_{j+1}$ at time $j+1$, but rather to *approximate* samples. This, however, does not affect the proposed methodology; it is blind to whether the given data forms exact or approximate samples. ♢

A major challenge in ensemble filtering and smoothing problems is that both the reference (forecast) and target (analysis) distributions for the map do not have analytical density functions; everything is defined through approximate ensembles, and hence combinations of Dirac measures. For this reason the transport approaches derived in Chapter 5, which rely on the explicit form of the reference density to define the loss function (for example, the KL divergence in (5.2)) do not apply.

In this section we will seek the map so that the push-forward distribution $T_\sharp\widehat{\pi}_{j+1}$ is close to the true filtering distribution $\pi_{j+1}$. We do this by minimizing the squared energy distance, noting that this distance measure is well-defined between two empirical measures. Recall from Definition 12.19 that this distance is

$$\begin{aligned}\mathsf{D}_{\mathrm{E}}^2(T_\sharp\widehat{\pi}_{j+1},\pi_{j+1}) = {} & 2\mathbb{E}^{(v,v')\sim T_\sharp\widehat{\pi}_{j+1}\otimes\pi_{j+1}}|v-v'| - \mathbb{E}^{(v,v')\sim T_\sharp\widehat{\pi}_{j+1}\otimes T_\sharp\widehat{\pi}_{j+1}}|v-v'| \\ & - \mathbb{E}^{(v,v')\sim \pi_{j+1}\otimes\pi_{j+1}}|v-v'|,\end{aligned} \tag{10.1a}$$

where we have suppressed the dependence of the map on the parameters $\theta$ and the observation $y^\dagger_{j+1}$ for conciseness. We note also that

$$\mathsf{D}_{\mathrm{E}}^2(T_\sharp\widehat{\pi}_{j+1},\pi_{j+1}) = 2\mathbb{E}^{(v,v')\sim \widehat{\pi}_{j+1}\otimes\pi_{j+1}}|T(v)-v'| - \mathbb{E}^{(v,v')\sim \widehat{\pi}_{j+1}\otimes\widehat{\pi}_{j+1}}|T(v)-T(v')| + c,$$

where $c$ is a constant that is independent of the transport $T$. Noting that we will optimize over (parameterized) transport $T$ we emphasize that this constant is hence irrelevant for our purposes here.

To evaluate the distance given only a forecast ensemble, we use importance sampling to write the expectation in the distance with respect to $\pi_{j+1}$ in terms of $\widehat{\pi}_{j+1}$ with the likelihood weights $w_{j,n} = \mathsf{l}(y^\dagger_{j+1}|\widehat{v}^{(n)}_{j+1})$. Given a forecast ensemble $\{\widehat{v}^{(n)}_{j+1}\}_{n=1}^N$, we then define the loss function for the transport parameters as

$$\mathsf{L}(\theta) = \frac{2}{N^2}\sum_{n,m=1}^N |T(\widehat{v}^{(n)}_{j+1};\theta) - \widehat{v}^{(m)}_{j+1}|w_{j,m} - \frac{1}{N^2}\sum_{n,m=1}^N |T(\widehat{v}^{(n)}_{j+1};\theta) - T(\widehat{v}^{(m)}_{j+1};\theta)|. \tag{10.2}$$

The following result shows that the optimal parameters

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{L}(\theta) \tag{10.3}$$

minimize the energy distance with certain arguments.

**Theorem 10.3.** *Given a forecast ensemble* $\{\widehat{v}^{(n)}_{j+1}\}_{n=1}^N$*, let*

$$\widehat{\pi}_{j+1} \coloneqq \sum_{n=1}^N \delta_{\widehat{v}^{(n)}_{j+1}}$$

*represent the forecast distribution at time $j+1$ and let*

$$\pi_{j+1}(v) \propto \mathsf{l}(y^\dagger_{j+1}|v)\widehat{\pi}_{j+1}(v)$$

*be the filtering distributions at time $j+1$, respectively. Then the optimal parameters for the transport map in (10.3) solve the problem*

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{D}_{\mathrm{E}}\Big(T(\cdot; y^\dagger_{j+1}, \theta)_\sharp \widehat{\pi}_{j+1}, \pi_{j+1}\Big).$$

*Proof.* For the empirical measure $\widehat{\pi}_{j+1}$, the filtering distribution is also an empirical measure and the energy distance $\mathsf{D}^2_{\mathrm{E}}(T(\cdot; y^\dagger_{j+1}, \theta)_\sharp \widehat{\pi}_{j+1}, \pi_{j+1})$ is given by the loss function $\mathsf{L}(\theta)$ up to an additive constant. Given that the energy distance is minimized when the measures are equal and the constant does not affect the minimizer, the optimal parameters coincide. □

Remark 10.4. While evaluating the loss function, and hence the learned transport map, relies on importance sampling, the forecast ensemble will be updated using the transport alone. The transport will push-forward an equally weighted ensemble of forecast to analysis samples. This may partially alleviate the degeneracy faced by particle filters where the ensemble eventually has an effective sample size of 1. Particle degeneracy arises from keeping the position of each particle fixed during the analysis step. This constraint forces particles to possibly remain in low probability under the filtering distribution after conditioning on an observation. Instead, the approach outlined in this section uses transport to move the particle positions in each analysis step. ♢

## 10.2 Learning Dependence of the Map on Data

In this section we again seek transports that map to the filtering distribution $\pi_{j+1}$. In this case, we let the map $T$ *explicitly* depend on both the predicted state $\widehat{v}_{j+1} \in \mathbb{R}^d$ and the true observation $y^\dagger_{j+1} \in \mathbb{R}^k$; in particular we try to learn the dependence of $T$ on not only the forecast $\widehat{v}_{j+1}$ but also the data $y^\dagger_{j+1}$. Our goal is to find the parameters $\theta \in \Theta \subseteq \mathbb{R}^p$ of the map $T$ so that, at least approximately,

$$\widehat{v}_{j+1} \sim \widehat{\pi}_{j+1} \Rightarrow T(\widehat{v}_{j+1}, y^\dagger_{j+1}; \theta) \sim \pi_{j+1}.$$

Here the filtering distribution $\pi_{j+1}$ is found from Bayes Theorem 1.2 with the prior coming from the forecast $\widehat{\pi}_{j+1}$ and the likelihood model $\mathsf{l}(y^\dagger_{j+1}|\cdot)$. Using the push-forward notation, or goal is to choose $\theta$ to get as close as possible to achieving

$$T(\cdot, y^\dagger_{j+1}; \theta)_\sharp \widehat{\pi}_{j+1} = \pi_{j+1}. \tag{10.4}$$

In this section we will aim to identify appropriate $\theta$ without likelihood evaluations, i.e., given only an ensemble from the joint distribution.

**Data Assumption 10.5.** *We are given pairs of i.i.d. samples $\{(\widehat{v}^{(n)}_{j+1}, y^{(n)}_{j+1})\}_{n=1}^N$ from the joint distribution $\gamma_{j+1}(v, y) \coloneqq \widehat{\pi}_{j+1}(v)\mathsf{l}(y|v)$.*

In the following two subsections we will present two formulations to find these maps. In Subsection 10.2.1 we will minimize an objective based on the energy distance, while Subsection 10.2.2 will leverage the transports learned via maximum likelihood in Chapter 5.

### 10.2.1 Minimizing the Energy Distance

Following the approach in Section 10.1, we will consider the objective to be the energy distance between the true and approximate filtering distributions in expectation over the observation $y^\dagger_{j+1}$ drawn from its marginal distribution $\kappa_{j+1}$. That is,

$$\begin{aligned}\mathbb{E}^{y^\dagger_{j+1}\sim\kappa_{j+1}}\left[\mathsf{D}^2_{\mathrm{E}}(T(\cdot,y^\dagger_{j+1})_\sharp\widehat{\pi}_{j+1},\pi_{j+1})\right] &= 2\mathbb{E}^{y^\dagger_{j+1}\sim\kappa_{j+1}}\mathbb{E}^{(v,v')\sim\widehat{\pi}_{j+1}\otimes\pi_{j+1}}|T(v,y^\dagger_{j+1})-v'|- \\ &\quad \mathbb{E}^{y^\dagger_{j+1}\sim\kappa_{j+1}}\mathbb{E}^{(v,v')\sim\widehat{\pi}_{j+1}\otimes\widehat{\pi}_{j+1}}|T(v,y^\dagger_{j+1})-T(v',y^\dagger_{j+1})|+c,\end{aligned}$$

where $c$ is again a constant that is independent of $T$. By using the factorization of the joint distribution $\gamma_{j+1}(v,y)=\pi_{j+1}(v)\kappa_{j+1}(y)$, we can write the expected energy distance as

$$\begin{aligned}\mathbb{E}^{y^\dagger_{j+1}\sim\kappa_{j+1}}\left[\mathsf{D}^2_{\mathrm{E}}(T(\cdot,y^\dagger_{j+1})_\sharp\widehat{\pi}_{j+1},\pi_{j+1})\right] &= 2\mathbb{E}^{(v,v',y^\dagger_{j+1})\sim\widehat{\pi}_{j+1}\otimes\gamma_{j+1}}|T(v,y^\dagger_{j+1})-v'| \\ &\quad -\mathbb{E}^{(v,v',y^\dagger_{j+1})\sim\widehat{\pi}_{j+1}\otimes\widehat{\pi}_{j+1}\otimes\kappa_{j+1}}|T(v,y^\dagger_{j+1})-T(v',y^\dagger_{j+1})|.\end{aligned}$$

This allows us to use the joint ensemble to define the following empirical loss function for the map parameters:

$$\mathsf{L}(\theta)=\frac{2}{N^2}\sum_{m,n=1}^{N}|T(\widehat{v}^{(n)}_{j+1},y^{(m)}_{j+1};\theta)-\widehat{v}^{(m)}_{j+1}|-\frac{1}{N^2}\sum_{l,m,n=1}^{N}|T(\widehat{v}^{(n)}_{j+1},y^{(l)}_{j+1};\theta)-T(\widehat{v}^{(m)}_{j+1},y^{(l)}_{j+1};\theta)|.$$

We then define the optimal parameters for the map as $\theta^\star\in\arg\min_{\theta\in\Theta}\mathsf{L}(\theta)$.

Remark 10.6. The two terms in the loss function $\mathsf{L}$ have opposing behavior. While minimizing the first term encourages the map $T(\cdot,y^{(m)}_{j+1})$ to push-forward the forecast ensemble to the state $v^{(m)}_{j+1}$ matching the observation $y^{(m)}_{j+1}$, the second term encourages diversity in the map evaluations. That is, the second term is maximized by increasing the difference between all pairwise map evaluations for each observation. ◇

Remark 10.7. The expectation in the second term of the loss function involves a tensor product of two measures for the state and observation. This expectation can be estimated from an ensemble of paired states and observations by permuting the pairings. That is, $(\widehat{v}^{(n)}_{j+1},y^{(l)}_{j+1})\sim\widehat{\pi}_{j+1}\otimes\kappa_{j+1}$ for $n\neq l$. ◇

### 10.2.2 Composed Maps Learned with Maximum Likelihood

In this subsection we consider an alternative approach to construct an observation-dependent transport. To do so, we first identify an invertible block-triangular map (see Assumption 6.7) that pushes forward the joint distribution $\gamma_{j+1}$ to a known reference density $\varrho$ using, for example, the maximum-likelihood approach introduced in Chapter 5. The following theorem shows how to use this map to construct a transport that pushes forward $\gamma_{j+1}$ to the filtering distribution $\pi_{j+1}$.

**Theorem 10.8.** *Let $S\colon \mathbb{R}^d \times \mathbb{R}^k \to \mathbb{R}^d$ be a triangular map of the form*

$$S(y, v) = \begin{bmatrix} y \\ S_2(y, v) \end{bmatrix}, \tag{10.5}$$

*where $S_2(y, \cdot)\colon \mathbb{R}^d \to \mathbb{R}^d$ is invertible for each $y \in \mathbb{R}^k$. We denote this inverse as $S_2(y, \cdot)^{-1}$. If $S$ pushes forward $\gamma_{j+1}(y, v)$ to a product reference $\varrho_1(y)\varrho_2(v)$, then*

$$T(y, v, y^{\dagger}_{j+1}) \coloneqq S_2(y^{\dagger}_{j+1}, \cdot)^{-1} \circ S_2(y, v) \tag{10.6}$$

*pushes forward $\gamma_{j+1}(y, v)$ to $\pi_{j+1}(v)$ for any $y^{\dagger}_{j+1} \in \mathbb{R}^k$.*

*Proof.* By Theorem 6.9, the second component $S_2(y, \cdot)$ of a triangular map of the form in (10.5) pushes forward the conditional $\mathbb{P}(v|y)$ of $\gamma_{j+1}(y, v)$ to $\varrho_2(v)$ for any $y \in \mathbb{R}^k$. Thus, the inverse map $S_2(y^{\dagger}_{j+1}, \cdot)^{-1}|_v$ pushes forward the conditional $\mathbb{P}(v|y^{\dagger}_{j+1})$ given by the filtering density $\pi_{j+1}$ to $\varrho_2$. Composing these maps gives us a transport pushing forward $\gamma$ to $\pi_{j+1}$. □

Given the joint ensemble of states and observations, we can define our analysis ensemble by building an estimator for the map $\widehat{S}_2$ using the procedure in Subsection 6.2.2 and evaluating the composed map at our joint ensemble:

$$v^{(n)}_{j+1} = \widehat{T}(y^{(n)}_{j+1}, \widehat{v}^{(n)}_{j+1}, y^{\dagger}_{j+1}) \coloneqq \widehat{S}_2(y^{\dagger}_{j+1}, \widehat{S}_2(\widehat{v}^{(n)}_{j+1}, \cdot)^{-1}|_{\widehat{v}^{(n)}_{j+1}}), \qquad n = 1, \dots, N.$$

Remark 10.9. The joint-to-posterior map depends on both the true observation $y^{\dagger}_{j+1}$ and synthetic observations $y_{j+1}$ at time $j+1$. For a fixed $y^{\dagger}_{j+1}$, the transformation from the forecast to analysis state can be seen as a stochastic transport due to the randomness in the synthetic observations $y_{j+1}$. In contrast, the transports in Sections 10.1 and (10.2.1) for a fixed observation are deterministic maps that push forward $\widehat{\pi}_{j+1}$ to $\pi_{j+1}$. ♢

While the framework above is quite general, in practice we may want to seek transports within a parameterized family of functions with a particular structure. The following result shows that these filters based on composed maps are related to classic ensemble Kalman filters.

**Theorem 10.10.** *Let $S$ be an invertible triangle map of the form in (10.5) where $S_2$ is the optimal map with respect to the loss function in (6.12) over the affine space*

$$\mathcal{S} = \{S_2(y, u) = A(u + By + c), A \in \mathbb{R}^{d \times d}, A \succ 0, B \in \mathbb{R}^{d \times k}, c \in \mathbb{R}^d\}.$$

*Then, the composed map in (10.6) has the form of the ensemble Kalman update*

$$T(y, v, y^{\dagger}_{j+1}) = v + \Sigma_{\widehat{v}_{j+1}, y_{j+1}} \Sigma^{-1}_{y_{j+1}, y_{j+1}} (y^{\dagger}_{j+1} - y),$$

*where $\Sigma_{\widehat{v}_{j+1}, y_{j+1}}$ and $\Sigma_{y_{j+1}, y_{j+1}}$ denote the cross-covariance of $(\widehat{v}_{j+1}, y_{j+1})$ and the covariance of $y_{j+1}$ under $\gamma_{j+1}$, respectively.*

*Proof.* From Theorem 6.20, minimizing the loss in (6.12) over the space of affine maps $\mathcal{S}$ yields the transformation

$$S_2(y, v) = \Sigma_{v|y}^{-1/2}\Big(v - \mathbb{E}[v] - \Sigma_{vy}\Sigma_{yy}^{-1}(y - \mathbb{E}[y])\Big),$$

where we have suppressed the dependence of the states in the covariance on time for conciseness. The inverse of the map evaluated at $y_{j+1}^\dagger$ is given by

$$S_2(y_{j+1}^\dagger, \cdot)^{-1}|_v = \Sigma_{v|y}^{1/2} v + \mathbb{E}[v] + \Sigma_{vy}\Sigma_{yy}^{-1}(y_{j+1}^\dagger - \mathbb{E}[y]).$$

Composing these maps cancels out $\Sigma_{v|y}^{-1/2}$ and the constant terms, yielding the desired result. □

## 10.3 Optimal Particle Filter Transport

In Section 7.1, we define sequential filtering by alternating two steps: prediction using the dynamics model $\mathsf{P}$, and analysis $\mathsf{A}$ by accounting for the realized observation. For notational convenience we define $\mathsf{A}_j(\cdot) = \mathsf{A}(\cdot; y_{j+1}^\dagger)$ where $\mathsf{A}$ is defined in (7.5). This can be summarized, similarly to Section 7.1, as

$$\pi_{j+1} = \mathsf{A}_j \circ \mathsf{P}\pi_j.$$

In this section we follow the strategy of the optimal particle filter (OPF) introduced in Section 7.8, which reverses these two steps. At each time step $j$, the OPF first applies an analysis step to sample the conditional distribution $\mathbb{P}(v_j|Y_j, y_{j+1}^\dagger)$ starting from $\mathbb{P}(v_j|Y_j)$; and second it applies a prediction step to sample from $\mathbb{P}(v_{j+1}|Y_j, y_{j+1}^\dagger)$. These two steps may be summarized using the relationship

$$\pi_{j+1} = \mathsf{P}_j^{\mathrm{OPF}} \circ \mathsf{A}_j^{\mathrm{OPF}}\pi_j.$$

While the OPF performs the analysis step using importance sampling, we will show in this section how to achieve the first step using transports. In particular, our goal is to find a map $T$ depending on some parameters $\theta$ so that

$$T(v_j, y_{j+1}^\dagger; \theta) \sim \mathsf{A}_j^{\mathrm{OPF}}\pi_j, \quad v_j \sim \pi_j. \tag{10.7}$$

In the previous two Sections 10.1 and 10.2 we showed how to build transports for conditioning given samples from the joint distribution of the forecast, under two data assumptions. In the first, Section 10.1, we assume that we can evaluate the likelihood of the observation $y_{j+1}^\dagger$, while in the second, Section 10.2 we require only samples drawn from the joint distribution of state and observation. We will now demonstrate how to apply both methods to define the transport in (10.7) in the following two subsections.

### 10.3.1 Learning with Likelihood Weights

Following the approach in Section 10.1, we first seek the transport in (10.7) that implicitly depends on the observation $y^\dagger_{j+1}$ by minimizing the energy distance

$$\theta^\star \in \arg\min_{\theta \in \Theta} \mathsf{D}_{\mathrm{E}}\Big(T(\cdot; y^\dagger_{j+1}, \theta)_\sharp \pi_j, \mathsf{A}_j^{\mathrm{OPF}} \pi_j\Big). \tag{10.8}$$

We make the following data assumption:

**Data Assumption 10.11.** *We are given i.i.d. samples* $\{v_j^{(n)}\}_{n=1}^N$ *from the filtering distribution* $\pi_j$ *at time* $j$ *and we are able to evaluate* $\mathbb{P}(y^\dagger_{j+1}|v_j^{(n)})$.

To compute expectations with respect to $\mathsf{A}_j^{\mathrm{OPF}}\pi_j$ in the objective, we similarly rely on an importance sampling step by sampling $v_j^{(n)}$ from $\pi_j$ and computing the likelihood weights for the observations $\mathbb{P}(y_{j+1}|v_j^{(n)})$. As shown in Section 7.8, in the setting (7.33) when the observation model is linear, the likelihood function for $y^\dagger_{j+1}$ given the state $v_j$ is a Gaussian with the closed-form

$$\mathcal{N}\Big(H\Psi(v_j), H\Sigma H^\top + \Gamma\Big).$$

We denote the evaluation at $y^\dagger_{j+1}$, given sample $v_j^{(n)}$, by $w_{j,n}$. Then, we can define a loss function for the map parameters, which is analogous to (10.2). Given an ensemble approximation $\{v_j^{(n)}\}_{n=1}^N \sim \pi_j$ to the analysis distribution at time $j$, we have the loss function

$$\mathsf{L}(\theta) = 2\frac{1}{N^2}\sum_{n,m=1}^N |T(v_j^{(n)};\theta) - v_j^{(m)}| w_{j,m} - \frac{1}{N^2}\sum_{n,m=1}^N |T(v_j^{(n)};\theta) - T(v_j^{(m)};\theta)|.$$

This is obtained from the energy distance formulation (10.8) after dropping $\theta-$independent terms. The optimal map parameters can then be computed by finding

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{L}(\theta).$$

The equivalence between minimizing $\mathsf{L}$ and the energy distance follows the same proof of Theorem 10.3.

After learning the map, we can approximately sample from $\mathbb{P}(v_j|Y_j, y^\dagger_{j+1})$ by evaluating the map $T(v_j^{(n)}; y^\dagger_{j+1}, \theta^\star)$ at each sample $v_j^{(n)} \sim \pi_j$. This process generates an ensemble from what is known as the *lag-1 smoothing distribution*, for the state given one future observation. Applying the Markov kernel given in (7.35) and (7.36), which defines the operator $\mathsf{P}_j^{\mathrm{OPF}}$, then generates samples from the analysis distribution $\pi_{j+1}$ starting from the smoothing ensemble.

### 10.3.2 Learning the Data Dependence

Following the approach in Section 10.2, working under Data Assumption 10.5, we now seek the transport in (10.7) that explicitly depends on the observation by minimizing the expected energy distance

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathbb{E}^{y_{j+1}^\dagger \sim \kappa_{j+1}} \Big[ \mathsf{D}_{\mathrm{E}}(T(\cdot; y_{j+1}^\dagger, \theta)_\sharp \pi_j, \mathsf{A}_j^{\mathrm{OPF}} \pi_j) \Big].$$

As compared to the previous subsection, we can implement this loss as long as we can sample from the joint distribution of $(v_j, y_{j+1})$ for the state and the observation at the next time. Given a collection of analysis samples $\{v_j^{(n)}\}_{n=1}^N \sim \pi_j$, we can sample synthetic observations by evaluating the dynamics model followed by the observation model for each sample. That is,

$$\begin{aligned} \widehat{v}_{j+1}^{(n)} &= \Psi(v_j^{(n)}) + \xi_j, \quad \xi_j \sim \mathcal{N}(0, \Sigma) \\ y_{j+1}^{(n)} &= h(\widehat{v}_{j+1}^{(n)}) + \eta_{j+1}^{(n)}, \quad \eta_{j+1}^{(n)} \sim \mathcal{N}(0, \Gamma). \end{aligned}$$

We then define the loss function for the map parameters as

$$\mathsf{L}(\theta) = \frac{2}{N^2} \sum_{m,n=1}^N |T(v_j^{(n)}, y_{j+1}^{(m)}; \theta) - v_j^{(m)}| - \frac{1}{N^2} \sum_{l,m,n=1}^N |T(v_j^{(n)}, y_{j+1}^{(l)}; \theta) - T(v_j^{(m)}, y_{j+1}^{(l)}; \theta)|,$$

where the optimal parameters are given by $\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{L}(\theta)$. After identifying the map parameters, we generate the ensemble from the lag-1 smoothing distribution $\mathsf{A}_j^{\mathrm{OPF}} \pi_j$ by evaluating the map $T(v_j^{(n)}, y_{j+1}^\dagger; \theta^\star)$ with the true observation at time $j+1$ at each sample $v_j^{(n)} \sim \pi_j$.

Remark 10.12. The procedure in Subsection 10.3.1 requires a closed-form expression for the likelihood weights. Thus it is only implementable for linear observation models and one step of filtering, as for the optimal particle filter, as explained in Section 7.8. On the other hand, the approach in this subsection can be implemented for general non-linear observation operators, as long as we can sample from the forecast dynamics and likelihood model. ♢

## 10.4 Bibliography

Learning approaches for ensemble filtering and smoothing based on transportation of measure have shown recent promise in generalizing classic Kalman approaches and reducing the error. The paper [294] introduces the approach of composing triangular transports as in Theorem 10.8 to build prior-to-posterior maps, while [4] seeks optimal transport maps by solving adversarial learning problems. A related framework known as Gaussian anamorphosis seeks invertible transformations of the state and observations where Kalman algorithms apply. See [123] for an approach based on nonlinear diagonal transformations and a generalization that uses invertible neural networks in [61]. Other nonlinear generalizations of ensemble Kalman filtering methods include [144, 10, 192]. A nonlinear filter that is derived using variational inference using transports that lie in reproducing kernel Hilbert spaces (see Example 12.13) is [256].

# Chapter 11

## Data Assimilation Using Learned Forecasting Models

In this chapter we consider having an ML forecast dynamics model $\Psi^a$ that approximates the true model $\Psi$; see Chapter 8 and Chapter 15 for ways to learn this model. The motivation for using $\Psi^a$ may be that $\Psi$ is unknown, or that the $\Psi^a$ is less computationally expensive to use for forecasting. Such a model (or multiple models) $\Psi^a$ can then be used for smoothing and filtering, or used in conjunction with $\Psi$ in ensemble-based algorithms to augment the ensemble size. Note that this is closely related to the use of surrogate models in inverse problems, discussed in Chapter 3. No explicit data assumptions are made in this chapter, as the surrogate models can potentially be learned from a variety of different data scenarios.

This chapter is laid out as follows. In Section 11.1 we state approximation properties of the smoothing and filtering distributions with a learned forecast model. In Section 11.2 we discuss multifidelity methods for making use of a learned forecast model, in addition to the original computational model, to improve data assimilation. In Section 11.3 we discuss multifidelity methods for estimating covariance matrices. Section 11.4 closes with bibliographical remarks.

## 11.1 Smoothing and Filtering Using Surrogate Models

### 11.1.1 Smoothing Problem

Recall definitions of $V^\dagger, Y^\dagger$ and $Y_j^\dagger$ from (7.3). From (7.8) we obtain the prior on $V^\dagger$ from the dynamics model, a probability density function in $\mathcal{P}(\mathbb{R}^{d(J+1)})$ given by

$$\rho(V^\dagger) \propto \exp\Bigl(-\frac{1}{2}|v_0^\dagger - m_0|_{C_0}^2 - \frac{1}{2}\sum_{j=0}^{J-1} |v_{j+1}^\dagger - \Psi(v_j^\dagger)|_\Sigma^2\Bigr). \tag{11.1}$$

Immediately after (7.8) we define

$$\eta^\dagger = \{\eta_1^\dagger, \dots, \eta_J^\dagger\} \in \mathbb{R}^{kJ},$$
$$\mathsf{h}(V^\dagger) = \{h(v_1^\dagger), \dots, h(v_J^\dagger)\},$$

where $\eta^\dagger \sim \mathcal{N}(0, \mathsf{\Gamma})$, and $\mathsf{\Gamma}$ is block diagonal with $\Gamma$ in each diagonal block. We then obtain the posterior

$$\Pi(V) \propto \mathsf{l}(Y^\dagger|V)\rho(V), \tag{11.2}$$

also a probability density function in $\mathcal{P}(\mathbb{R}^{d(J+1)})$, where

$$\mathsf{l}(Y^\dagger|V) \propto \exp\Bigl(-\frac{1}{2}|Y^\dagger - \mathsf{h}(V)|_\Gamma^2\Bigr). \tag{11.3}$$

(We drop the superscript $Y^\dagger$ on the posterior as it is not central to the following discussion.) If we use a surrogate model $\Psi^a$ to accelerate computations we will change the prior dynamics model to have the form

$$\rho'(V^\dagger) \propto \exp\Bigl(-\frac{1}{2}|v_0^\dagger - m_0|_{C_0}^2 - \frac{1}{2}\sum_{j=0}^{J-1}|v_{j+1}^\dagger - \Psi^a(v_j^\dagger)|_\Sigma^2\Bigr). \tag{11.4}$$

The resulting posterior is

$$\Pi'(V) \propto \mathsf{l}(Y^\dagger|V)\rho'(V), \tag{11.5}$$

We are interested in what effect the use of a surrogate model has on the posterior. As a first step we simply assume that $\rho'$ is close to $\rho$ and ask what can be said about the closeness of $\Pi'$ and $\Pi$. In the following theorem and proof, all integrals are over $\mathbb{R}^{d(J+1)}$.

**Theorem 11.1.** *Assume that $Z := \int \mathsf{l}(Y^\dagger|V)\rho(V)\,dV > 0$. Then, there is a constant $C > 0$ such that, for all $\mathsf{D}_{\mathrm{TV}}(\rho,\rho')$ sufficiently small,*

$$\mathsf{D}_{\mathrm{TV}}(\Pi,\Pi') \le C\mathsf{D}_{\mathrm{TV}}(\rho,\rho').$$

*Proof.* In this proof $Z' := \int \mathsf{l}(Y^\dagger|V)\rho'(V)\,dV > 0$ and $\mathsf{l}(V) = \mathsf{l}(Y^\dagger|V)$. We have

$$\begin{aligned} 2\mathsf{D}_{\mathrm{TV}}(\Pi,\Pi') &= \int\Bigl|\frac{1}{Z}\mathsf{l}(V)\rho(V) - \frac{1}{Z'}\mathsf{l}(V)\rho'(V)\Bigr|\,dV \\ &\le I_1 + I_2, \\ I_1 &:= \int\Bigl|\frac{1}{Z}\mathsf{l}(V)\rho(V) - \frac{1}{Z}\mathsf{l}(V)\rho'(V)\Bigr|\,dV, \\ I_2 &:= \int\Bigl|\frac{1}{Z}\mathsf{l}(V)\rho'(V) - \frac{1}{Z'}\mathsf{l}(V)\rho'(V)\Bigr|\,dV. \end{aligned}$$

Since $Z > 0$ and $\mathsf{l}(V)$ is uniformly bounded with respect to $V$ we deduce that, for some $C_1 > 0$,

$$I_1 \le C_1\mathsf{D}_{\mathrm{TV}}(\rho,\rho').$$

It also follows that, for some $C_2 > 0$,

$$|Z - Z'| \le \int \mathsf{l}(V)|\rho(V) - \rho'(V)|\,dV \le C_2\mathsf{D}_{\mathrm{TV}}(\rho,\rho').$$

Hence, for all $\mathsf{D}_{\mathrm{TV}}(\rho,\rho')$ sufficiently small, $Z' > \frac{1}{2}Z > 0$. Finally we deduce that, for some $C_1 > 0$, for some $C_3 > 0$,

$$I_2 = \Bigl|\frac{1}{Z} - \frac{1}{Z'}\Bigr|Z \le C_3|Z - Z'|.$$

The result follows. □

Remark 11.2. The sense in which $\rho'$ is close to $\rho$ depends on details of surrogate models, out of distribution, that we do not get into in these notes. In particular the approximation theorems we allude to in Section 14.4 are valid, in their simplest form, over compact sets $D$; in contrast, the prior distribution here is supported on the whole Euclidean space. However, by making assumptions about the behaviour of $\Psi$ and $\Psi^a$ at infinity, which controls errors outside $D$, and by choosing $D$ large enough, it is possible to deduce that $\rho'$ is close to $\rho$. ♢

### 11.1.2 3DVar

Consider the filtering problem of estimating signal $\{v_j\}_{j\geq 0}$ given observations $\{y_j\}_{j\geq 1}$ in the setting where the dynamics-observation models (7.1), (7.2) reduce to

$$\begin{aligned} v^{\dagger}_{j+1} &= \Psi(v^{\dagger}_j),\\ y^{\dagger}_{j+1} &= Hv^{\dagger}_{j+1} + \eta^{\dagger}_{j+1}. \end{aligned}$$

We make the same assumptions on the noise and initial conditions as detailed at the start of Chapter 7. We are interested in applications where evaluating $\Psi$ is computationally expensive, but where we have a surrogate, i.e., an approximate model $\Psi^a$ that can be cheaply evaluated. The theorem below proves long-time accuracy for a filtering algorithm that uses the surrogate dynamics $\Psi^a$ rather than the true dynamics $\Psi$.

The following theorem proves long-time accuracy for a 3DVar filtering algorithm that uses the surrogate dynamics $\Psi^a$ rather than the true dynamics $\Psi$. The result relies on standard observability conditions on the true dynamics and observation model $(\Psi, H)$ and on accuracy of the surrogate model $\Psi^a$ in the unobserved part of the state-space. Such a result may have some interest because we do not assume long-time accuracy or stability of the surrogate dynamics, but we can nevertheless obtain long-time accuracy of filtering estimates by leveraging the observations $\{y^{\dagger}_j\}_{j\geq 1}$.

**Theorem 11.3.** *Let Assumption 7.22 hold and let $K$ be a gain matrix appearing in the mean update of the 3DVar method with a surrogate forecast model $\Psi^a$. That is,*

$$m_{j+1} = (I - KH)\Psi^a(m_j) + Ky^{\dagger}_{j+1}.$$

*Assume there exists constants $\lambda \in (0,1)$ and $\delta < \infty$ so that*

$$\begin{aligned} \sup_{v\in\mathbb{R}^d} |(I - KH)D\Psi(v)| &\leq \lambda,\\ \sup_{v\in\mathbb{R}^d} \bigl|(I - KH)\bigl(\Psi(v) - \Psi^a(v)\bigr)\bigr| &= \delta, \end{aligned}$$

*and denote*

$$\epsilon = \mathbb{E}|K\eta_j|.$$

*Then, the 3DVar estimate based on the biased forecast model satisfies*

$$\limsup_{j\to\infty} \mathbb{E}|m_j - v^{\dagger}_j| = \frac{\epsilon + \delta}{1 - \lambda}.$$

*Proof.* This is simply Theorem 7.23. □

## 11.2 Multifidelity Ensemble State Estimation

In this section we assume that we have an ensemble of traditional numerical solvers, but want to reduce the error variance in state estimation given an ensemble of ML forecasts. In particular, learned forecast models often have lower computational cost than numerical ones, but may be less accurate. Even if they are less accurate, the forecasts from the learned forecast models can often be used to improve ensemble-based DA by increasing the ensemble size of the numerical model.

However, when the models have different accuracy, they must be treated differently in the DA process. We consider the numerical model to be the computationally expensive, high-fidelity model, and the (possibly multiple) learned forecast models to be the computationally cheap, low-fidelity models. Multi-model and multifidelity methods can be applied in this context; we discuss two such methods below. The first is the multi-model EnKF, which makes the statistical assumption that all the models are unbiased. The second is an EnKF based on multifidelity Monte Carlo, which does not assume unbiasedness, but maintains unbiasedness with respect to the high-fidelity model. Our discussion includes analysis of how to pick the ensemble sizes optimally, balancing cost and accuracy within a given computational budget.

### 11.2.1 Multi-Model (Ensemble) Kalman Filters

#### Linear Gaussian Setting

We work in the setting where the dynamics-observations models (7.1) and (7.2) are linear and Gaussian and take the form

$$v^\dagger_{j+1} = Av^\dagger_j + \xi^\dagger_j, \tag{11.6a}$$

$$y^\dagger_{j+1} = Hv^\dagger_{j+1} + \eta^\dagger_{j+1}. \tag{11.6b}$$

We again make the same assumptions on the noise and initial conditions as detailed at the start of Chapter 7. We introduce projection matrices $G_l : \mathbb{R}^d \mapsto \mathbb{R}^{d_l}$. If we assume there is $A_l$ such that $G_l A = A_l G_l$, and define $\Sigma_l = G_l \Sigma G_l^\top$, then we obtain

$$G_l v^\dagger_{j+1} = A_l G_l v^\dagger_j + \varepsilon_{l,j}, \tag{11.7}$$

where $\varepsilon_{l,j} \sim \mathcal{N}(0, \Sigma_l)$. If we define $v^\dagger_{l,j} = G_l v^\dagger_j$ then we have

$$v^\dagger_{l,j+1} = A_l v^\dagger_{l,j} + \varepsilon_{l,j}. \tag{11.8}$$

We may now ask how we can use the multiple models (11.8) to develop a Kalman filter for (11.6). An answer is given by the *multi-model Kalman filter* which is defined as follows:

$$\widehat{v}_{l,j+1} = A_l v_{l,j}, \tag{11.9a}$$
$$\widehat{C}_{l,j+1} = A_l C_{l,j} A_l^\top + \Sigma_l, \tag{11.9b}$$
$$C_{j+1} = \left( \sum_{l=1}^{L} G_l^\top \widehat{C}_{l,j+1}^{-1} G_l + H^\top \Gamma^{-1} H \right)^{-1}, \tag{11.9c}$$
$$v_{j+1} = C_{j+1} \left( \sum_{l=1}^{L} G_l^\top \widehat{C}_{l,j+1}^{-1} \widehat{v}_{l,j+1} + H^\top \Gamma^{-1} y_{j+1}^\dagger \right), \tag{11.9d}$$
$$v_{l,j+1} = G_l v_{j+1}, \tag{11.9e}$$
$$C_{l,j+1} = G_l C_{j+1} G_l^\top. \tag{11.9f}$$

It can be readily seen from (11.9c) that if $G_l$ is injective and $\widehat{C}_{l,j+1}$ is positive definite, then model $l$ decreases the covariance of the multi-model analysis $v_{j+1}$. The multi-model Kalman filter can be derived in a Bayesian way, or as the best linear unbiased estimator (BLUE) for multiple models. Indeed, following the latter route, the analysis step of the MM-KF follows from the following theorem.

**Theorem 11.4.** *Assume that $\{\widehat{v}_l\}_{l=1}^L$ are defined from $v^\dagger \in \mathbb{R}^d$ by the identities*

$$\widehat{v}_l = G_l v^\dagger + e_l,$$

*where $\mathbb{E}[e_l] = 0$, $\mathbb{E}[e_l e_l^\top] = C_l$, and $\mathbb{E}[e_l e_{l'}^\top] = 0$ when $l \neq l'$.*

*These estimates may be combined to obtain the minimum variance linear unbiased estimator of $v^\dagger$, which is given by*

$$\widehat{v} = \sum_{l=1}^{L} B_l \widehat{v}_l,$$

*where*

$$B_l = \left( \sum_{l'=1}^{L} G_{l'}^\top C_{l'}^{-1} G_{l'} \right)^{-1} G_l^\top C_l^{-1}.$$

*Proof.* Note that we are seeking the BLUE estimator for a linear inverse problem

$$y = \overline{G} v^\dagger + e,$$

where $y = (\widehat{v}_1, \ldots, \widehat{v}_L) \in \mathbb{R}^{dL}$, $e = (e_1, \ldots, e_L)$, the covariance of $e$ is $\overline{C}$, and

$$\overline{C} = \begin{pmatrix} C_1 & & \\ & \ddots & \\ & & C_L \end{pmatrix}, \qquad \overline{G} = \begin{pmatrix} G_1 \\ \vdots \\ G_L \end{pmatrix}.$$

By the Gauss–Markov theorem, the solution is given by the OLS estimator

$$(\overline{G}^\top \overline{C}^{-1} \overline{G})^{-1} \overline{G}^\top \overline{C}^{-1} y,$$

which agrees with the desired expression.

□

### Nonlinear Non-Gaussian Setting

We now generalize (11.6), (11.8) to the nonlinear and non-Gaussian setting, assuming that

$$v^\dagger_{j+1} = \Psi(v^\dagger_j) + \xi^\dagger_j, \tag{11.10a}$$

$$y^\dagger_{j+1} = Hv^\dagger_{j+1} + \eta^\dagger_{j+1}, \tag{11.10b}$$

and that we have multiple nonlinear forward models $\{\Psi_l\}_{l=1}^L$ so that

$$v^\dagger_{l,j+1} = \Psi_l(v^\dagger_{l,j}) + \varepsilon_{l,j}, \tag{11.11}$$

where $v^\dagger_{l,j} = G_l v^\dagger_j$. We make the same assumptions on the noise and initial conditions as in the linear Gaussian case.

Similar to the single-model EnKF, if instead of linear models $A_l$ we have nonlinear models $\Psi_l$, we can formally replace the forecast step involving linear models with the nonlinear ones:

$$\widehat{v}^{(n)}_{l,j+1} = \Psi_l(v^{(n)}_{l,j}) + \xi^{(n)}_{l,j}, \quad n = 1, \ldots, N_l, \tag{11.12}$$

and replace the means and covariance matrices in (11.9) with their empirical versions estimated from the ensembles.

The equations (11.9) can also be written in an iterative form, in which the analysis step of the single-model Kalman filter is repeatedly applied to each new model forecast. Other multi-model Kalman filters can then be obtained by replacing the Kalman filter analysis step with one of its variants, such as the EnKF. The multi-model EnKF (MM-EnKF) can also make use of nonlinear $G_l$ and $H$. We will not state the details, but point the reader to the Bibliography for further information.

### 11.2.2 Multifidelity Monte Carlo

Here we work in the setting of the preceding Subsection 11.2.1, in the nonlinear non-Gaussian context. An alternative approach to what is presented in the previous subsection is one based on control variates. This does not make the assumption of unbiasedness, but does require a known hierarchy of fidelities.

The idea of multifidelity Monte Carlo methods comes from the concept of *control variates*, a way of reducing the variance in Monte Carlo estimates. The basic idea is to use a correlated random variable, which may have a different expectation, to reduce the variance in the primary.

Suppose again that we have $L$ low-fidelity models $\{\Psi_l\}_{l=1}^L$.[1] Assume that $\Psi_{\text{hi}}$ is the high-fidelity model (corresponding to $l = 0$), with respect to which we would like to be unbiased. Assume furthermore that the ensemble sizes for each model are arranged such that $0 < N_{\text{hi}} \le N_1 \le \ldots \le N_L$. We deal with the scalar case for simplicity, and also recall that we are working in the linear observation setting where $h(\cdot) = H\cdot$.

[1] For notational simplicity, we do not write a $\Sigma_l$ for each model. However, the results in this subsection apply equally well if $\Psi_l$ is considered to be the stochastic function $\Psi'_l(v) = \Psi_l(v) + \xi$, where $\xi \sim \mathcal{N}(0, \Sigma_l)$.

Let $\{v^{(n)}\}_{n=1}^{N_L}$ be i.i.d. samples from density $\Upsilon$. Then compute the following ensemble means:

$$\widehat{m}_{\mathrm{hi}} = \frac{1}{N_{\mathrm{hi}}} \sum_{n=1}^{N_{\mathrm{hi}}} \Psi_{\mathrm{hi}}(v^{(n)}), \qquad \widehat{m}_l = \frac{1}{N_l} \sum_{n=1}^{N_l} \Psi_l(v^{(n)}),$$
$$\widehat{m}_l' = \frac{1}{N_{l-1}} \sum_{n=1}^{N_{l-1}} \Psi_l(v^{(n)}).$$

Then we consider a multifidelity estimator given by

$$\widehat{m} = \widehat{m}_{\mathrm{hi}} + \sum_{l=1}^{L} \alpha_l (\widehat{m}_l - \widehat{m}_l'), \tag{11.13}$$
$$\alpha_l = \frac{\rho_l \sigma_{\mathrm{hi}}}{\sigma_l}, \tag{11.14}$$

where $\rho_l$ is the correlation coefficient between $\Psi_{\mathrm{hi}}$ and $\Psi_l$, and $\sigma_l^2$, $\sigma_{\mathrm{hi}}^2$ are the variances of $\Psi_l$ and $\Psi_{\mathrm{hi}}$, respectively.

**Theorem 11.5.** *The estimator $\widehat{m}$ is unbiased for $\mathbb{E}^{v\sim\Upsilon}\left[\Psi_{\mathrm{hi}}(v)\right]$ and has variance*

$$\mathrm{Var}[\widehat{m}] = \frac{\sigma_{\mathrm{hi}}^2}{N_{\mathrm{hi}}} + \sum_{l=1}^{L} \Big( N_{l-1}^{-1} - N_l^{-1} \Big) (\alpha_l^2 \sigma_l^2 - 2\alpha_l \rho_l \sigma_{\mathrm{hi}} \sigma_l). \tag{11.15}$$

*The variance* $\mathrm{Var}[\widehat{m}]$ *is minimized when*

$$\alpha_l = \frac{\rho_l \sigma_{\mathrm{hi}}}{\sigma_l}.$$

*Proof.* Note that expectation here is with respect to $\Upsilon$, the distribution of the samples $\{v^{(n)}\}_{n=1}^{N_L}$. The fact that $\mathbb{E}[\widehat{m}] = \mathbb{E}[\widehat{m}_{\mathrm{hi}}] = \mathbb{E}[\Psi_{\mathrm{hi}}]$ is immediate from the linearity of expectation.

Then,

$$\begin{aligned}
\mathrm{Var}[\widehat{m}] = \mathrm{Var}[\widehat{m}_{\mathrm{hi}}] &+ \sum_{l=1}^{L} \alpha_l^2 (\mathrm{Var}[\widehat{m}_l] + \mathrm{Var}[\widehat{m}_l']) \\
&+ 2\sum_{l=1}^{L} \alpha_l (\mathrm{Cov}[\widehat{m}_{\mathrm{hi}}, \widehat{m}_l] - \mathrm{Cov}[\widehat{m}_{\mathrm{hi}}, \widehat{m}_l']) \\
&+ 2\sum_{l=1}^{L} \alpha_l \sum_{j=l+1}^{L} \alpha_j (\mathrm{Cov}[\widehat{m}_l, \widehat{m}_j] - \mathrm{Cov}[\widehat{m}_l, \widehat{m}_j']) \\
&- 2\sum_{l=1}^{L} \alpha_l \sum_{j=l+1}^{L} \alpha_j (\mathrm{Cov}[\widehat{m}_l', \widehat{m}_j] - \mathrm{Cov}[\widehat{m}_l', \widehat{m}_j']) \\
&- 2\sum_{l=1}^{L} \alpha_l^2 \mathrm{Cov}[\widehat{m}_l, \widehat{m}_l'].
\end{aligned}$$

Lines 3 and 4 cancel. Using the definitions of the correlation coefficients and variances, and the fact that the variance of a mean of $N$ independent random variables decreases as $1/N$, we obtain

$$\mathrm{Var}[\widehat{m}] = \frac{\sigma_{\mathrm{hi}}^2}{N_{\mathrm{hi}}} + \sum_{l=1}^{L} \Big( N_{l-1}^{-1} - N_l^{-1} \Big) (\alpha_l^2 \sigma_l^2 - 2\alpha_l \rho_l \sigma_{\mathrm{hi}} \sigma_l).$$

Taking the derivative of $\mathrm{Var}[\widehat{m}]$ with respect to $\alpha_l$ and setting it to 0 gives the stated expression for $\alpha_l$. □

If $\{\sigma_l\}_{l=1}^L$, $\sigma_{\mathrm{hi}}$, and $\{\rho_l\}_{l=1}^L$ are not known, they can be estimated from the ensembles. This suggests the procedure outlined in Algorithm 11.1.

**Algorithm 11.1** Ensemble Kalman filter using multifidelity Monte Carlo

```
1: Input: Initialization $(v_0, C_0)$.
2: for $j = 1, \dots, J$ do
3:     Sample $\{v_j^{(n)}\}_{n=1}^{N_M} \sim \mathcal{N}(m_{j-1}, C_{j-1})$.
4:     Evaluate the models at the $v_j^{(n)}$ and compute the means $m_{\mathrm{hi},j}$, $m_{l,j}$ and $m'_{l,j}$ for
       $l = 1, \dots, L$.
5:     Estimate $\sigma_{\mathrm{hi},j}$, $\{\sigma_{l,j}\}_{l=1}^L$, and $\{\rho_{l,j}\}_{l=1}^L$ from the ensembles.
6:     Compute the multifidelity Monte Carlo estimate of the mean $\widehat{m}_j$ (11.13) and its
       associated variance $\mathrm{Var}[\widehat{m}_j]$ (11.15). Then, estimate the forecast covariance $\widehat{C}'_j$
       from the ensembles, and set
           $\widehat{C}_j = \widehat{C}'_j + \mathrm{Var}[\widehat{m}_j]$.
       See the remark below on the computation of $\widehat{C}'_j$.
7:     Assimilate observations, updating the mean and covariance:
           $m_j = \widehat{m}_j + K_j(y_j^\dagger - H\widehat{m}_j)$,
           $C_j = (I - K_j H)\widehat{C}_j$,
           $K_j = \widehat{C}_j H^\top (H\widehat{C}_j H^\top + \Gamma)^{-1}$.
8: end for
9: Output: $(v_J, C_J)$.
```

Remark 11.6. We have thus far discussed multifidelity estimation of the mean of a distribution. In step 6 of Algorithm 11.1, the forecast covariance $\widehat{C}'_j$ can itself be estimated using a multifidelity method, as discussed further in Section 11.3 and in the bibliography. Alternatively, this covariance can be estimated from the ensemble of one of the fidelity levels. ♢

### 11.2.3 Model Selection and Sample Allocation

For a fixed computational budget, we would like to find the allocation of ensemble sizes $\{N_l\}_{l=1}^L$ that minimizes the variance while staying within the budget. This is called the *model selection and sampling allocation problem* (MOSAP).

We apply this idea to the multifidelity Monte Carlo estimator (11.13). Assume the cost of a single run of model $l$ is $c_l$, and we have a fixed computational budget $b$. We would like to minimize the variance (11.15) under this constraint, leading to

$$\begin{aligned} &\min_{\{N_l\}_{l=0}^L} \frac{\sigma_{\text{hi}}^2}{N_{\text{hi}}} + \sum_{l=1}^L \Big(N_{l-1}^{-1} - N_l^{-1}\Big)(\alpha_l^2\sigma_l^2 - 2\alpha_l\rho_l\sigma_{\text{hi}}\sigma_l) \\ &\text{subject to } \sum_{l=1}^L c_l N_l = b,\ N_{l-1} \le N_l. \end{aligned}$$

This is an integer programming problem. Similar analyses can be carried out for other multifidelity methods.

## 11.3 Multifidelity Covariance Estimation

Thus far, we have discussed multifidelity methods for improving state estimates. Within ensemble DA, due to the effect of sampling error, it is also of interest to improve estimates of forecast covariance matrices, and to possibly mitigate the need for localization and inflation as discussed in Subsections 7.6.3 and 7.6.2. This can be done by using a large ML ensemble to improve the estimated covariance produced by a small numerical ensemble. Alternatively, an ML model can be trained to predict the covariance directly.

Consider having access to multiple unbiased estimators $\{\widehat{C}_l\}_{l=1}^L$ of a covariance matrix $\widehat{C}$, and taking a scalar linear combination of these estimators:

$$\sum_{l=1}^L c_l \widehat{C}_l, \tag{11.17}$$

If each of the $\widehat{C}_l$ is symmetric positive definite, then the linear combination will be too. Under Gaussian assumptions on these estimators, the following theorem gives the improvement in the covariance estimate.

**Theorem 11.7.** *Suppose that we have $L$ independent estimators of the true covariance $\widehat{C}$, $\{\widehat{C}_l\}_{l=1}^L$. Assume furthermore that they are all unbiased, and are distributed according to $\widehat{C}_l \sim \mathcal{N}(\widehat{C}, U_l, V_l)$, the matrix normal distribution (note that we do not make use of the fact that $\widehat{C}_l$ will generally be positive semidefinite).*

*Consider a linear combination of $\widehat{C}_l$ with scalar weights,*

$$\sum_{l=1}^L c_l \widehat{C}_l,$$

*such that $\sum_{l=1}^L c_l = 1$ (this is necessary for the combination to be unbiased).*

*Then, the* $\{c_l\}_{l=1}^L$ *that minimize the total variance of* $\sum_{l=1}^L c_l \widehat{C}_l$ *are*

$$c_l = \frac{e_l^{-1}}{\sum_{j=1}^L e_j^{-1}}, \tag{11.18}$$

*where* $e_l = \mathbb{E}\Big[|\widehat{C}_l - \widehat{C}|_F^2\Big]$ *and* $|\cdot|_F$ *is the Frobenius norm. The total variance is then*

$$\left(\sum_{l=1}^L e_l^{-1}\right)^{-1}. \tag{11.19}$$

*Proof.* In squared Frobenius norm, $\widehat{C}_l$ has an expected error of

$$\begin{aligned} e_l &= \mathbb{E}\Big[|\widehat{C}_l - \widehat{C}|_F^2\Big] \\ &= \mathbb{E}\Big[\mathrm{Tr}\Big((\widehat{C}_l - \widehat{C})^\top(\widehat{C}_l - \widehat{C})\Big)\Big] \\ &= \mathrm{Tr}\Big(\mathbb{E}\Big[(\widehat{C}_l - \widehat{C})^\top(\widehat{C}_l - \widehat{C})\Big]\Big) \\ &= \mathrm{Tr}(U_l)\mathrm{Tr}(V_l), \end{aligned}$$

where the last line follows from a property of the matrix normal distribution.

Then, for the linear combination (11.17),

$$\mathrm{vec}\left(\sum_{l=1}^L c_l \widehat{C}_l\right) \sim \mathcal{N}\left(\mathrm{vec}\left(\sum_{l=1}^L c_l \widehat{C}\right), \sum_{l=1}^L c_l^2 (V_l \otimes U_l)\right) = \mathcal{N}\left(\mathrm{vec}(\widehat{C}), \sum_{l=1}^L c_l^2 (V_l \otimes U_l)\right),$$

where $\otimes$ is the Kronecker product. Now, the total variance is

$$\mathrm{Tr}\left(\sum_{l=1}^L c_l^2 (V_l \otimes U_l)\right) = \sum_{l=1}^L c_l^2 \mathrm{Tr}(V_l \otimes U_l) = \sum_{l=1}^L c_l^2 \mathrm{Tr}(U_l)\mathrm{Tr}(V_l) = \sum_{l=1}^L c_l^2 e_l. \tag{11.20}$$

Minimizing (11.20) with respect to the $c_l$ and using Lagrange multipliers to enforce $\sum_{l=1}^L c_l = 1$, we obtain

$$c_l = \frac{e_l^{-1}}{\sum_{j=1}^L e_j^{-1}}.$$

Note that the $c_l$ are positive. The total variance becomes

$$\left(\sum_{l=1}^L e_l^{-1}\right)^{-1}.$$

□

Remark 11.8. Note that if the $\widehat{C}_l$ are sample covariance matrices from an ensemble, and the ensemble members are normally distributed, the $\widehat{C}_l$ will have a Wishart distribution. In the large-ensemble limit the Wishart sampling distribution tends towards a matrix normal distribution. ♢

The bibliography points to some multifidelity covariance estimators that do not require the assumption of unbiasedness.

## 11.4 Bibliography

Data assimilation using learned forecast models has been considered in [127, 195, 52, 248, 337]. The importance of these forecast models correctly reproducing the Lyapunov spectrum and forecast error covariance was explored in [248]. Including the Lyapunov spectrum and attractor dimension into the training process was explored in [251]. Instead of learning a full forward model, learning an adjoint model for use in 4DVar was considered in [136].

Hybrid methods combining a numerical and learned forecast model in data assimilation have been considered in [18, 53]. The latter used a large ensemble of a learned forecast model, along with a smaller ensemble of a more expensive numerical model solving the equations of motion, in an EnKF, mitigating the need for localization. Combining a small high-fidelity ensemble with a large ensemble of reduced-order models in an EnKF using control variates was considered in [253]. A multilevel EnKF was introduced in [145].

The multi-model Kalman filter was introduced in [228], with a Bayesian derivation in the latter, and a best linear unbiased estimator (BLUE) derivation provided in [17]. The multi-model EnKF was introduced in [17]. Additionally, there have been ML methods for learning to combine multiple model forecasts [310, 134]. These have not been used with data assimilation to our knowledge.

The generation of ensembles from an ML forecast model was considered in [283, 196, 254].

A survey of multifidelity methods is presented in [247]. The multifidelity Monte Carlo estimator discussed in this chapter was introduced in [246], including the optimal ensemble sizes for each model balancing computational cost and accuracy. We refer to [124] for background on the matrix normal distribution and its properties. A framework for analyzing different multifidelity estimators and finding the multi-level best linear unbiased estimator (MLBLUE) is presented in [282], and later generalized to a multivariate setting in [68]. Multifidelity estimation of covariance matrices was considered in [79, 214, 215].

The relation between the matrix normal distribution and the Wishart distribution is discussed in [229].

# Part III

# Learning Frameworks

# Chapter 12

## Metrics, Divergences and Scoring Rules

In this chapter we define various "distance-like" ways to quantify closeness between probability measures. We work with probability measures on $\mathbb{R}^d$, denoted $\mathcal{P}(\mathbb{R}^d)$. We also discuss scoring rules which quantify closeness of a probability measure to a point, and deterministic scoring rules, between two points. To be consistent with the rest of the notes, the presentation focuses on probability density functions; but the definitions all work for general probability distributions. Indeed we will, in several instances, employ Dirac masses. Generically we compare two probability density functions $\rho$ and $\varrho$; we denote the cumulative density functions associated with these two probability density functions by $F_\rho$ and $F_\varrho$. We start with metrics in Section 12.1; we continue by discussing divergences in Section 12.2; and finally we discuss scoring rules in Section 12.3. Throughout the chapter we will make connections between metrics, divergences and scoring rules, as summarized in Figure 12.1.

Our motivation for studying this topic is twofold. Some metrics and divergences are useful to present theoretical results about inverse problems and data assimilation. Others are useful for defining loss functions for probabilistic machine learning models based on data, which is important for designing machine learning algorithms and assessing their performance. For example, the Hellinger metric is useful for stating stability results for measures; see, for example, Theorems 1.11 and 3.4. On the other hand, the energy distance is a metric that is used as a loss function for tasks such as learning probabilistic filters; see, for example, Section 10.1. As well as being useful for stating theoretical results, the Kullback–Leibler divergence is useful for defining loss functions in machine learning; see Chapter 5. Scoring rules have been used traditionally for evaluating probabilistic forecasts against samples and hold potential to define objectives for machine learning tasks.

## 12.1 Metrics

### 12.1.1 Metrics on the Space of Probability Measures

A metric between probability distributions is a function $\mathsf{D}\colon \mathcal{P}(\mathbb{R}^d) \times \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}$ that satisfies the following four properties for all $\rho, \varrho \in \mathcal{P}(\mathbb{R}^d)$:

1. Non-negative: $\mathsf{D}(\rho, \varrho) \geq 0$.
2. Positive: $\mathsf{D}(\rho, \varrho) = 0$ if and only if $\rho = \varrho$.

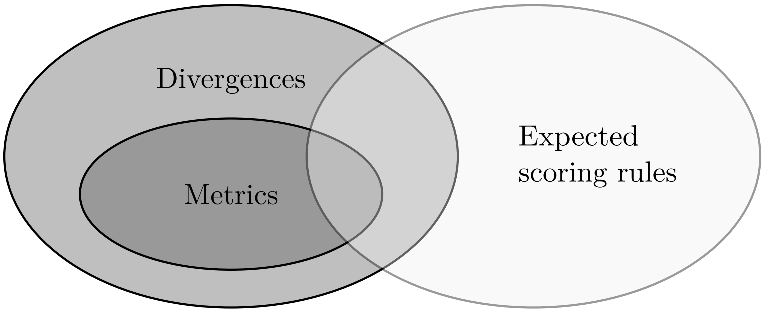

**Figure 12.1** Diagram representing the constraints of three groups of distance-like ways to quantify closeness between probability measures. Metrics impose more restrictive conditions than divergences: only some divergences are metrics. Furthermore a subset of expected scoring rules, those based on strictly proper rules lead to divergences, and in some cases to metrics. We provide concrete examples of such scoring rules in the following sections.

3. Symmetric: $\mathsf{D}(\rho, \varrho) = \mathsf{D}(\varrho, \rho)$.
4. Sub-additive or triangle inequality: $\mathsf{D}(\rho, \varrho) \leq \mathsf{D}(\rho, \pi) + \mathsf{D}(\pi, \varrho)$ for all $\pi \in \mathcal{P}(\mathbb{R}^d)$.

### 12.1.2 Total Variation and Hellinger Metrics

**Definition 12.1.** The *total variation distance* between two probability density functions $\rho$ and $\varrho$ is defined by[1]

$$\mathsf{D}_{\mathrm{TV}}(\rho, \varrho) \coloneqq \frac{1}{2} \int |\rho(u) - \varrho(u)| \, du = \frac{1}{2} \|\rho - \varrho\|_{L^1}.$$

The *Hellinger distance* between two probability density functions $\rho$ and $\varrho$ is defined by

$$\mathsf{D}_{\mathrm{H}}(\rho, \varrho) \coloneqq \left( \frac{1}{2} \int \left| \sqrt{\rho(u)} - \sqrt{\varrho(u)} \right|^2 du \right)^{1/2} = \frac{1}{\sqrt{2}} \|\sqrt{\rho} - \sqrt{\varrho}\|_{L^2}.$$

$\diamondsuit$

We now establish bounds between the Hellinger and total variation distance.

**Lemma 12.2.** *For any probability density functions $\rho$ and $\varrho$:*

- *The total variation and Hellinger metrics are uniformly bounded; indeed*

$$0 \leq \mathsf{D}_{\mathrm{TV}}(\rho, \varrho) \leq 1, \quad 0 \leq \mathsf{D}_{\mathrm{H}}(\rho, \varrho) \leq 1;$$

- *The total variation and Hellinger metrics bound one another; indeed*

$$\frac{1}{\sqrt{2}} \mathsf{D}_{\mathrm{TV}}(\rho, \varrho) \leq \mathsf{D}_{\mathrm{H}}(\rho, \varrho) \leq \sqrt{\mathsf{D}_{\mathrm{TV}}(\rho, \varrho)}.$$

[1] Here the integrals are over $\mathbb{R}^d$. For simplicity, in this chapter we do not write the domain of integration when it is clear by context.

*Proof.* For part (i) we note that the lower bounds follow immediately from the definitions, so we only need to prove the upper bounds. For total variation distance

$$\mathsf{D}_{\mathrm{TV}}(\rho,\varrho)=\frac{1}{2}\int|\rho(u)-\varrho(u)|\,du\le\frac{1}{2}\int\rho(u)\,du+\frac{1}{2}\int\varrho(u)\,du=1,$$

and for Hellinger distance

$$\begin{aligned}\mathsf{D}_{\mathrm{H}}(\rho,\varrho)&=\left(\frac{1}{2}\int\left|\sqrt{\rho(u)}-\sqrt{\varrho(u)}\right|^2du\right)^{1/2}\\&=\left(\frac{1}{2}\int\left(\rho(u)+\varrho(u)-2\sqrt{\rho(u)\varrho(u)}\,\right)du\right)^{1/2}\\&\le\left(\frac{1}{2}\int\left(\rho(u)+\varrho(u)\right)du\right)^{1/2}\\&=1.\end{aligned}$$

For part (ii) we use the Cauchy-–Schwarz inequality

$$\begin{aligned}\mathsf{D}_{\mathrm{TV}}(\rho,\varrho)&=\frac{1}{2}\int\left|\sqrt{\rho(u)}-\sqrt{\varrho(u)}\right|\left|\sqrt{\rho(u)}+\sqrt{\varrho(u)}\right|du\\&\le\left(\frac{1}{2}\int\left|\sqrt{\rho(u)}-\sqrt{\varrho(u)}\right|^2du\right)^{1/2}\left(\frac{1}{2}\int\left|\sqrt{\rho(u)}+\sqrt{\varrho(u)}\right|^2du\right)^{1/2}\\&\le\mathsf{D}_{\mathrm{H}}(\rho,\varrho)\left(\frac{1}{2}\int\left(2\rho(u)+2\varrho(u)\right)du\right)^{1/2}\\&=\sqrt{2}\mathsf{D}_{\mathrm{H}}(\rho,\varrho).\end{aligned}$$

Notice that $|\sqrt{\rho(u)}-\sqrt{\varrho(u)}|\le|\sqrt{\rho(u)}+\sqrt{\varrho(u)}|$ since $\sqrt{\rho(u)},\sqrt{\varrho(u)}\ge0$. Thus we have

$$\begin{aligned}\mathsf{D}_{\mathrm{H}}(\rho,\varrho)&=\left(\frac{1}{2}\int\left|\sqrt{\rho(u)}-\sqrt{\varrho(u)}\right|^2du\right)^{1/2}\\&\le\left(\frac{1}{2}\int\left|\sqrt{\rho(u)}-\sqrt{\varrho(u)}\right|\left|\sqrt{\rho(u)}+\sqrt{\varrho(u)}\right|du\right)^{1/2}\\&\le\left(\frac{1}{2}\int|\rho(u)-\varrho(u)|\,du\right)^{1/2}\\&=\sqrt{\mathsf{D}_{\mathrm{TV}}(\rho,\varrho)}.\end{aligned}$$

□

Remark 12.3. The preceding proof may be extended to show that $\rho$ and $\varrho$ have total variation and Hellinger distance equal to one if and only if $\int\rho(u)\varrho(u)\,du=0$; that is, if and only if they have disjoint supports. ♢

In addition to relating closeness of densities to closeness of expectations with respect to different densities, the following lemma also provides a useful characterization of the total variation distance.

**Lemma 12.4.** *Let $f : \mathbb{R}^d \to \mathbb{R}$. If two densities are close in total variation or in Hellinger distance, expectations computed with respect to both densities are also close.*

- *(i) Let $f$ be a function such that $|f|_\infty := \sup_{u \in \mathbb{R}^d} |f(u)| < \infty$. It holds that*

  $$\bigl|\mathbb{E}^\rho[f] - \mathbb{E}^\varrho[f]\bigr| \le 2|f|_\infty \mathsf{D}_{\mathrm{TV}}(\rho, \varrho).$$

  *In fact, the following variational characterization of the total variation distance holds:*

  $$\mathsf{D}_{\mathrm{TV}}(\rho, \varrho) = \frac{1}{2} \sup_{|f|_\infty \le 1} \bigl|\mathbb{E}^\rho[f] - \mathbb{E}^\varrho[f]\bigr|. \tag{12.1}$$

- *(ii) Let $f$ be a function such that $f_2 := \bigl(\mathbb{E}^\rho[|f|^2] + \mathbb{E}^\varrho[|f|^2]\bigr)^{1/2} < \infty$. It holds that*

  $$\bigl|\, \mathbb{E}^\rho[f] - \mathbb{E}^\varrho[f]\bigr| \le 2 f_2 \mathsf{D}_{\mathrm{H}}(\rho, \varrho).$$

*Proof.* For part (i) we start by noting that

$$\begin{aligned}
\bigl|\mathbb{E}^\rho[f] - \mathbb{E}^\varrho[f]\bigr| &= \Bigl| \int f(u)\bigl(\rho(u) - \varrho(u)\bigr)\, du \Bigr| \\
&\le 2|f|_\infty \cdot \frac{1}{2} \int |\rho(u) - \varrho(u)|\, du \\
&= 2|f|_\infty \mathsf{D}_{\mathrm{TV}}(\rho, \varrho).
\end{aligned}$$

For any $f$ with $|f|_\infty = 1$ we obtain

$$\mathsf{D}_{\mathrm{TV}}(\rho, \varrho) \ge \frac{1}{2} \bigl|\, \mathbb{E}^\rho[f] - \mathbb{E}^\varrho[f]\bigr|.$$

We complete the proof of part (i) by exhibiting a choice of $f$ with $|f|_\infty = 1$ that achieves equality. To this end we choose $f(u) := \mathrm{sign}\Bigl(\rho(u) - \varrho(u)\Bigr)$, so that $f(u)\Bigl(\rho(u) - \varrho(u)\Bigr) = |\rho(u) - \varrho(u)|$. Then $|f|_\infty = 1$, and

$$\begin{aligned}
\mathsf{D}_{\mathrm{TV}}(\rho, \varrho) &= \frac{1}{2} \int |\rho(u) - \varrho(u)|\, du \\
&= \frac{1}{2} \int f(u)\Bigl(\rho(u) - \varrho(u)\Bigr)\, du \\
&= \frac{1}{2} \bigl|\mathbb{E}^\rho[f] - \mathbb{E}^\varrho[f]\bigr|.
\end{aligned}$$

For part (ii) of the lemma we may use the Cauchy–Schwarz inequality to show that

$$
\begin{aligned}
\left|\mathbb{E}^{\rho}[f]-\mathbb{E}^{\varrho}[f]\right| &= \left|\int f(u)\Big(\sqrt{\rho(u)}-\sqrt{\varrho(u)}\Big)\Big(\sqrt{\rho(u)}+\sqrt{\varrho(u)}\,\Big)\,du\right| \\
&\le \left(\frac{1}{2}\int\Big|\sqrt{\rho(u)}-\sqrt{\varrho(u)}\Big|^2\,du\right)^{1/2}\left(2\int|f(u)|^2\Big|\sqrt{\rho(u)}+\sqrt{\varrho(u)}\Big|^2\,du\right)^{1/2} \\
&\le \mathsf{D}_{\mathrm{H}}(\rho,\varrho)\Big(4\int|f(u)|^2(\rho(u)+\varrho(u))\,du\Big)^{1/2} \\
&= 2f_2\,\mathsf{D}_{\mathrm{H}}(\rho,\varrho).
\end{aligned}
$$

□

### 12.1.3 Transportation Metrics

#### Kantorovich Formulation

Given two densities $\rho$ and $\varrho$ on $\mathbb{R}^d$, we define a *coupling* as a density $\pi$ on $\mathbb{R}^d\times\mathbb{R}^d$ with the property that

$$\int_{\mathbb{R}^d}\pi(z,u)\,du=\rho(z),\quad \int_{\mathbb{R}^d}\pi(z,u)\,dz=\varrho(u).$$

Thus the marginals of $\pi$ deliver $\rho$ and $\varrho$. We denote the set of all such couplings by $\Pi_{\rho,\varrho}$. Given a cost function $\mathsf{c}:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}^+$ we define the Kantorovich formulation of the optimal transport problem as follows:

$$\pi^\star=\mathrm{arginf}_{\pi\in\Pi_{\rho,\varrho}}\int_{\mathbb{R}^d\times\mathbb{R}^d}\mathsf{c}(z,u)\pi(z,u)\,dzdu. \tag{12.2}$$

This notion of optimal transport may be used to define families of metrics on probability densities as follows.

**Definition 12.5.** Given a metric $\mathsf{d}(\cdot,\cdot)$ on $\mathbb{R}^d$ and an integer $p\ge 1$, the *Wasserstein–$p$ distance* between two probability density functions $\rho$ and $\varrho$ is defined by

$$W_p(\rho,\varrho):=\left(\inf_{\pi\in\Pi_{\rho,\varrho}}\int_{\mathbb{R}^d\times\mathbb{R}^d}\mathsf{d}(z,u)^p\pi(z,u)\,dzdu\right)^{1/p}. \tag{12.3}$$

♢

This metric on probability measures has two key aspects: first, it relates the metric to an underlying metric on $\mathbb{R}^d$; and second, it allows for a meaningful distance to be calculated between measures which are mutually singular.

**Example 12.6.** Let $\rho=\mathcal{N}(\mu_\rho,\Sigma_\rho)$ and $\varrho=\mathcal{N}(\mu_\varrho,\Sigma_\varrho)$. It may be shown that, if the metric $\mathsf{d}$ induced by the Euclidean norm $|\cdot|$ is used, then

$$W_2(\rho,\varrho)^2=|\mu_\rho-\mu_\varrho|^2+\mathrm{Tr}\Big(\Sigma_\rho+\Sigma_\varrho-2\big(\Sigma_\rho^{\frac{1}{2}}\Sigma_\varrho\Sigma_\rho^{\frac{1}{2}}\big)\Big).$$

In particular, if $\varrho$ is a Dirac measure at the origin then

$$W_2(\rho,\varrho)^2 = |\mu_\rho|^2 + \mathrm{Tr}(\Sigma_\rho).$$

Thus, in the Wasserstein$-2$ metric Gaussian $\rho$ is close to a Dirac at the origin if the mean and covariance of $\rho$ are both small. In contrast, the total variation distance between Gaussian $\rho$ and a Dirac at the origin is maximal, and equal to 1, unless the two measures coincide $(\mu_\rho = 0, \Sigma_\rho = 0)$ when it is 0.

Notice also that if $\rho$ and $\varrho$ are Diracs at $\mu_\rho$ and $\mu_\varrho$, then $W_2(\rho,\varrho) = |\mu_\rho - \mu_\varrho|$; thus closeness of the mass locations $\mu_\rho$ and $\mu_\varrho$ in Euclidean space translates into closeness of $\rho$ and $\varrho$ in Wasserstein distance. ♢

### Monge Formulation

The connection of optimal transport to explicit transport maps is made clear in the Monge formulation of optimal transport. In this formulation we explicitly link $\varrho$ with a pushforward of $\rho$ through a transport map $g$ by identifying the map which minimizes the cost:

$$g^\star = \mathrm{arginf}_{g: g_\sharp\rho=\varrho} \int_{\mathbb{R}^d} \mathsf{c}\bigl(z, g(z)\bigr)\rho(z)\,dz. \tag{12.4}$$

Under certain smoothness assumptions, the Kantorovich formulation has solution within the Monge class. That is, the optimal coupling is constructed using the pushforward of a transport map:

$$\pi^\star(z,u) = \delta\bigl(u - g^\star(z)\bigr)\rho(z). \tag{12.5}$$

Remark 12.7. For the Euclidean norm metric $\mathsf{d}(z,u) = |z-u|$, and $\rho$ being absolutely continuous with respect to the Lebesgue measure (that is it has a probability density function), there exists an optimal transport coupling in (12.3) solving the Kantorivich formulation of the Wasserstein-$p$ optimal transport problem. Moreover, this coupling has the form in (12.5), which is induced by an optimal transport map $g^\star$ solving the corresponding Monge problem (12.4). We note that for $p > 1$ the objective in (12.3) is strictly convex, and so the optimal transport map that solves $W_p(\rho,\varrho)$ is also unique; however this result does not extend to $p = 1$. ♢

### One-Dimensional Setting

The optimal transport metric has as an advantage that it directly links a metric on $\mathbb{R}^d$ to the metric on $\mathcal{P}(\mathbb{R}^d)$. However it is an implicit definition, via an optimization, and it can be hard to gain insight about it. In one dimension, however, there are a number of explicit formulae for the optimal transport metric which are insightful and which we describe here.

**Lemma 12.8.** *Let $\rho,\varrho$ be the densities of two real-valued random variables with invertible cumulative distribution functions $F_\rho, F_\varrho \colon \mathbb{R} \to [0,1]$. Then, for $p \ge 1$ the Wasserstein-$p$ distance with metric $\mathsf{d}$ induced by the Euclidean norm $|\cdot|$ has the form*

$$W_p(\rho,\varrho)^p = \int_0^1 |F_\rho^{-1}(q) - F_\varrho^{-1}(q)|^p\,dq.$$

*Proof.* The map $g(z) = F_\varrho^{-1} \circ F_\rho(z)$ pushes forward $\rho$ to $\varrho$, and it can be shown to be optimal in the one-dimensional setting. By substituting this map in the objective and performing the change of variables $q = F_\rho(z)$, we have

$$\begin{aligned} W_p(\rho, \varrho)^p &= \inf_{g_\sharp \rho = \varrho} \int_{\mathbb{R}} |z - g(z)|^p \rho(z)\, dz \\ &= \int_{\mathbb{R}} |z - F_\varrho^{-1} \circ F_\rho(z)|^p \rho(z)\, dz \\ &= \int_0^1 |F_\rho^{-1}(q) - F_\varrho^{-1}(q)|^p \, dq. \end{aligned}$$

□

The Wasserstein-$p$ distance for $p = 1$ has some additional computational advantages. In addition to being an integral probability metric, as discussed in Example 12.12, the closed-form expression in one dimension can be computed without requiring the inverse of cumulative distribution functions.

**Lemma 12.9.** *In the setting of Lemma 12.8, the Wasserstein-1 distance can be expressed as*

$$W_1(\rho, \varrho) = \int_0^1 |F_\rho^{-1}(q) - F_\varrho^{-1}(q)|\, dq = \int_{\mathbb{R}} |F_\rho(z) - F_\varrho(z)|\, dz.$$

*Proof.* The proof of this result is shown using Figure 12.2. It illustrates that integrating the difference between the cumulative distribution functions vertically (left) is equivalent to the horizontal integration (right). □

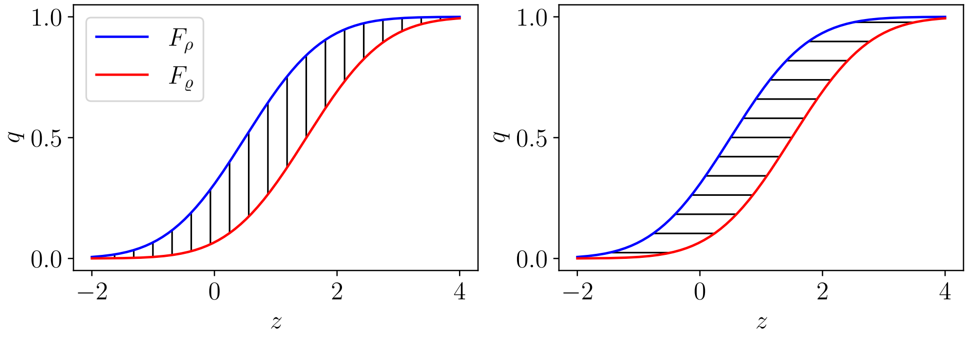

**Figure 12.2** Integration of the difference between cumulative distribution functions (left) and their inverses (right)

### 12.1.4 Integral Probability Metrics

We now build on the variational characterization (12.1) of the total variation distance. Let $\mathcal{F}$ be a set of real-valued functions $f\colon \mathbb{R}^d \to \mathbb{R}$ known as discriminators. These are used to distinguish two probability distributions according to the maximum difference of expectations of functions from $\mathcal{F}$: for each discriminator $f \in \mathcal{F}$, we compute the expectation of this function under the two distributions and define a metric by maximizing the difference of the expectations between the distributions.

**Definition 12.10.** Let $\mathcal{F}$ be a set of discriminator functions. An *integral probability metric* (IPM) between densities $\rho, \varrho$ is defined by

$$\mathsf{D}_{\mathcal{F}}(\rho, \varrho) := \sup_{f \in \mathcal{F}} \big| \, \mathbb{E}^{\rho}[f] - \mathbb{E}^{\varrho}[f] \big|. \tag{12.6}$$

♢

Notice that $\mathsf{D}_{\mathcal{F}}$ is a metric if and only if $\mathcal{F}$ separates points on the space of probability distributions: for any two different probability measures $\rho, \varrho$ there is a point in $\mathcal{F}$ with different expectations under $\rho$ and $\varrho$.

**Example 12.11.** Perhaps the most fundamental example of an IPM is the total variation distance, given in (12.1). For this metric, $\mathcal{F}$ is the scaled unit ball of functions bounded by $\frac{1}{2}$ in $L^\infty$. ♢

**Example 12.12.** A second important example of an IPM is the Wasserstein-1 distance where $\mathcal{F}$ is the space of Lipschitz continuous functions with Lipschitz constant less than or equal to one. ♢

**Example 12.13.** A third commonly occurring example of an IPM arises when $\mathcal{F}$ is defined as the unit ball of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ of functions $f : D \subseteq \mathbb{R}^d \to \mathbb{R}$. As discussed in Subsection 14.3.1, associated with $\mathcal{H}$ there is a symmetric and non-negative kernel $c : D \times D \to \mathbb{R}$ with the property that pointwise evaluation of a function $f \in \mathcal{H}$ can be computed by the inner product

$$f(u) = \langle f, c(u, \cdot) \rangle_{\mathcal{H}}.$$

This is known as the *reproducing property.* The next subsection is devoted to IPMs defined through an RKHS. Lemma 12.15 shows that, in this case, the supremum in (12.6) is given in closed form in terms of evaluations of the kernel. The resulting metric is known as the maximum mean discrepancy. ♢

### 12.1.5 Maximum Mean Discrepancy and Energy Distance

**Definition 12.14.** Let $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ be a symmetric and non-negative kernel. The *maximum mean discrepancy* (MMD) between two probability density functions $\rho$ and $\varrho$ is defined by

$$\mathsf{D}^2_{\mathrm{MMD}}(\rho, \varrho) := \mathbb{E}^{(u,u') \sim \rho \otimes \rho}\big[c(u, u')\big] + \mathbb{E}^{(v,v') \sim \varrho \otimes \varrho}\big[c(v, v')\big] - 2\mathbb{E}^{(u,v) \sim \rho \otimes \varrho}\big[c(u, v)\big].$$

♢

**Lemma 12.15.** *Define $\mathcal{F} = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq 1\}$, the unit ball of an RKHS $\mathcal{H}$ with reproducing kernel $c$. Then the MMD is an IPM with set of discriminator functions $\mathcal{F}$.*

*Proof.* Let $m_\rho(\cdot) = \mathbb{E}^{u\sim\rho}[c(u,\cdot)]$ denote the mean embedding of a distribution $\rho$. By the reproducing property for functions $f$ in a RKHS $\mathcal{H}$ we have

$$\mathbb{E}^\rho[f] = \int f(u)\rho(u)\,du = \int \langle f, c(u,\cdot)\rangle_{\mathcal{H}}\rho(u)\,du = \left\langle f, \int c(u,\cdot)\rho(u)\,du \right\rangle_{\mathcal{H}} = \langle f, m_\rho\rangle_{\mathcal{H}}.$$

Using the dual definition of the norm $\|g\|_{\mathcal{H}} = \sup_{f\in\mathcal{F}}\langle f, g\rangle_{\mathcal{H}}$, we have

$$\begin{aligned}\sup_{f\in\mathcal{F}}|\mathbb{E}^\rho[f] - \mathbb{E}^\varrho[f]| &= \sup_{f\in\mathcal{F}}|\langle f, m_\rho - m_\varrho\rangle_{\mathcal{H}}| \\ &= \|m_\rho - m_\varrho\|_{\mathcal{H}} \\ &= \Big(\langle m_\rho, m_\rho\rangle_{\mathcal{H}} - 2\langle m_\rho, m_\varrho\rangle_{\mathcal{H}} + \langle m_\varrho, m_\varrho\rangle_{\mathcal{H}}\Big)^{1/2}.\end{aligned}$$

The right-hand side agrees with $\mathsf{D}_{\text{MMD}}(\rho,\varrho)$. To see this, we can apply for each term the reproducing property with $k$ to compute the inner product of the mean embedding. For example,

$$\begin{aligned}\langle m_\rho, m_\varrho\rangle_{\mathcal{H}} &= \Big\langle \mathbb{E}^{u\sim\rho}[c(u,\cdot)], \mathbb{E}^{u'\sim\varrho}[c(u',\cdot)]\Big\rangle_{\mathcal{H}} \\ &= \mathbb{E}^{(u,u')\sim\rho\otimes\varrho}\Big[\langle c(u,\cdot), c(u',\cdot)\rangle_{\mathcal{H}}\Big] \\ &= \mathbb{E}^{(u,u')\sim\rho\otimes\varrho}\big[c(u,u')\big]. \qquad (12.7)\end{aligned}$$

□

Remark 12.16. The reproducing property in (12.7) permits the computation of inner products with respect to the kernel (or an associated infinite-dimensional feature map) using evaluations of the kernel alone. In many machine learning tasks, this property is referred to as the *kernel trick.* ♢

Remark 12.17. The MMD is a metric under certain conditions on the kernel. In particular, we say that the kernel is characteristic if $\mathbb{E}^\rho[f] = \mathbb{E}^\varrho[f]$ for all $f$ in the RKHS defined by $c(\cdot,\cdot)$ implies that $\rho = \varrho$. In order words, characteristic kernels produce metrics that satisfy the positivity condition. ♢

**Example 12.18.** Gaussian and Laplace kernels are examples of characteristic kernels. Choosing $c(u,u') = -|u-u'|$, the negative of the Euclidean distance, produces the energy distance, which we now define. It is also a metric, despite corresponding to a negative kernel. ♢

**Definition 12.19.** The *energy distance* $\mathsf{D}_{\text{E}}$ between two probability density functions $\rho$ and $\varrho$ is defined by

$$\mathsf{D}_{\text{E}}^2(\rho,\varrho) := 2\mathbb{E}^{(u,v)\sim\rho\otimes\varrho}|u-v| - \mathbb{E}^{(u,u')\sim\rho\otimes\rho}|u-u'| - \mathbb{E}^{(v,v')\sim\varrho\otimes\varrho}|v-v'|. \qquad (12.8)$$

♢

Proof of the following is postponed to the end of Subsection 12.3.1 where we discuss energy scores:

**Lemma 12.20.** *The energy distance* $\mathsf{D}_{\mathrm{E}}$ *is a metric on* $\mathcal{P}(\mathbb{R}^d)$.

A key property of the MMD and the energy distance is that they are amenable to ensemble approximation; they can be implemented without explicit formulae for either of $\rho$ or $\varrho$. For example, given independent samples $\{u^i\}_{i=1}^N \sim \rho$ and $\{v^i\}_{i=1}^M \sim \varrho$, we can estimate the MMD distance by

$$\widehat{\mathsf{D}^2_{\mathrm{MMD}}}(\rho,\varrho) = \frac{1}{N(N-1)}\sum_{i\neq j} c(u^i,u^j) + \frac{1}{M(M-1)}\sum_{i\neq j} c(v^i,v^j) - \frac{2}{NM}\sum_{i,j} c(u^i,v^j). \tag{12.9}$$

Likewise, we can estimate the energy distance by

$$\widehat{\mathsf{D}^2_{\mathrm{E}}}(\rho,\varrho) = \frac{2}{NM}\sum_{i,j}|u^i - v^j| - \frac{1}{N(N-1)}\sum_{i\neq j}|u^i-u^j| - \frac{1}{M(M-1)}\sum_{i\neq j}|v^i-v^j|. \tag{12.10}$$

Remark 12.21. Another important property of both the MMD with isotropic kernel and the energy distance is that they are rotation invariant. That is, the metrics do not change under the transformations $u \leftarrow Ru$ and $v \leftarrow Rv$ for some unitary matrix $R$. ♢

### 12.1.6 Metrics on the Space of Random Probability Measures

The space of random probability measures arises naturally when building methodologies from empirical sampling or from Monte Carlo methods. We consider functions $\pi : \Omega \to \mathcal{P}(\mathbb{R}^d)$, for some abstract probability space $(\Omega, \mathcal{B}, \mathbb{P})$; expectation under $\mathbb{P}$ is denoted by $\mathbb{E}$. For any fixed $\omega \in \Omega$ let $\pi = \pi(\omega)$ and define $\pi(f) := \int f(u)\pi(u)\,du$.

**Definition 12.22.** We define the following metric on the space of random probability measures:

$$d(\pi,\pi') := \sup_{|f|_\infty \le 1}\left|\mathbb{E}\Big[\big(\pi(f) - \pi'(f)\big)^2\Big]\right|^{1/2}.$$

♢

Remark 12.23. We note that $d$ reduces to the total variation distance (up to a 1/2 factor) when $\pi, \pi'$ are not random. ♢

We now define Monte Carlo approximation of $\pi$:

$$\pi^N_{\mathrm{MC}}(f) = \frac{1}{N}\sum_{n=1}^N f(u^{(n)}),\; u^{(n)} \sim \pi \quad \text{i.i.d.} \tag{12.11}$$

The metric as defined is well-adapted to measure the error in approximating $\pi$ by $\pi^N_{\mathrm{MC}}$ :

**Theorem 12.24.** *Given* $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ *define* $|f|_\infty := \sup_{u\in\mathbb{R}^d}|f(u)|$. *Then*

$$\sup_{|f|_\infty\le 1}\left|\mathbb{E}\left[\pi^N_{\mathrm{MC}}(f) - \pi(f)\right]\right| = 0,$$

$$d(\pi^N_{\mathrm{MC}},\pi)^2 \le \frac{1}{N}.$$

*Proof.* To prove that the estimator is unbiased, use linearity of the expected value and that $u^{(n)} \sim \pi$ to obtain the following identity, from which the desired result follows:

$$\begin{aligned}\mathbb{E}\Big[\pi^N_{\mathrm{MC}}(f)\Big] &= \mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N} f\big(u^{(n)}\big)\right]\\ &= \frac{1}{N} N\pi(f) = \pi(f) = \mathbb{E}\big[\pi(f)\big].\end{aligned}$$

For the error estimate in $d(\cdot,\cdot)$ we note that, since $\pi^N_{\mathrm{MC}}(f)$ is unbiased, its variance coincides with its mean squared error. Using the fact that the $u^{(n)} \sim \pi$ are independent we deduce that

$$\begin{aligned}\mathrm{Var}\Big[\pi^N_{\mathrm{MC}}(f)\Big] &= \mathrm{Var}\left[\frac{1}{N}\sum_{n=1}^{N} f\big(u^{(n)}\big)\right]\\ &= \frac{1}{N^2} N \mathrm{Var}_\pi[f] = \frac{1}{N}\mathrm{Var}_\pi[f].\end{aligned}$$

Assuming $|f|_\infty \le 1$, we have

$$\mathrm{Var}_\pi[f] = \pi(f^2) - \pi(f)^2 \le \pi(f^2) \le 1,$$

and therefore

$$\sup_{|f|_\infty \le 1}\left|\mathbb{E}\left[\Big(\pi^N_{\mathrm{MC}}(f) - \pi(f)\Big)^2\right]\right| = \sup_{|f|_\infty \le 1}\left|\frac{1}{N}\mathrm{Var}_\pi[f]\right| \le \frac{1}{N}.$$

□

## 12.2 Divergences

Recall the conditions for a metric given in Section 12.1. If we remove the conditions of symmetry and the triangle inequality then we obtain a *statistical divergence.* This distance-like measure between two probability distributions is defined as a function $\mathsf{D}\colon \mathcal{P}(\mathbb{R}^d) \times \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}$ that only satisfies the non-negativity and positivity conditions for a metric. To emphasize the asymmetry in their arguments, divergences are often written with double bars $\mathsf{D}(\rho\|\varrho)$ or a colon $\mathsf{D}(\rho : \varrho)$. We adopt the former convention.

### 12.2.1 $\mathsf{f}$-Divergences

An important feature of $\mathsf{f}$-divergences, which we now define, is that they depend on a ratio of two probability density functions. To simplify the presentation, throughout this section we restrict our attention to positive densities.

**Definition 12.25.** Let $\mathsf{f}\colon [0,\infty) \to \mathbb{R}$ be a convex function that satisfies $\mathsf{f}(0) = \lim_{t\to 0^+} \mathsf{f}(t)$ and $\mathsf{f}(1) = 0$. The $\mathsf{f}$-*divergence* between distributions with positive densities $\rho$ and $\varrho$ is defined by

$$\mathsf{D}_{\mathsf{f}}(\rho\|\varrho) := \int \mathsf{f}\left(\frac{\rho(u)}{\varrho(u)}\right)\varrho(u)\,du.$$

♢

Remark 12.26. Some $\mathsf{f}$-divergences define metrics. For example, the choice $\mathsf{f}(t) = \frac{1}{2}|t-1|$ delivers $\mathsf{D}_\mathsf{f}(\rho\|\varrho) = \mathsf{D}_{\mathrm{TV}}(\rho, \varrho)$ and the choice $\mathsf{f}(t) = (1-\sqrt{t})^2$ delivers $\mathsf{D}_\mathsf{f}(\rho\|\varrho) = 2\mathsf{D}_{\mathrm{H}}(\rho, \varrho)^2$. It can be shown that the only $\mathsf{f}$-divergence that is also an IPM is the total variation distance. The next definition introduces two important $\mathsf{f}$-divergences that do not define metrics: the Kullback-Leibler divergence and the $\chi^2$ divergence. They correspond to choosing $\mathsf{f}(t) = t\log t$ and $\mathsf{f}(t) = (t-1)^2$, respectively. ♢

**Definition 12.27.** The *Kullback–Leibler (KL) divergence* between two positive probability density functions $\rho$ and $\varrho$ is defined by

$$\mathsf{D}_{\mathrm{KL}}(\rho\|\varrho) := \int \log\left(\frac{\rho(u)}{\varrho(u)}\right)\rho(u)\,du. \tag{12.12}$$

The $\chi^2$ *divergence* between two positive probability density functions $\rho$ and $\varrho$ is defined by

$$\mathsf{D}_{\chi^2}(\rho\|\varrho) := \int \left(\frac{\rho(u)}{\varrho(u)} - 1\right)^2 \varrho(u)\,du. \tag{12.13}$$

♢

The $\chi^2$ divergence arises naturally in the analysis of sampling algorithms that involve weighting samples according to the ratio between two densities. In that context, the $\chi^2$ divergence has a natural interpretation as quantifying the variance of the weights. The KL divergence has a distinguished place among $\mathsf{f}$-divergences due to its appealing analytical and computational properties. The following remark shows that KL can be minimized over its first argument without knowledge of the normalizing constant of the second, which is useful in variational inference (see Chapter 2).

Remark 12.28. For any $\mathsf{f}$-divergence it holds that

$$\pi \in \arg\min_{q\in\mathcal{P}} \mathsf{D}_\mathsf{f}(q\|\pi)$$

and the minimizer is unique. The KL divergence is the only $\mathsf{f}$-divergence with the property that, for any $c > 0$, $\mathsf{D}_\mathsf{f}(q\|c\pi) - \mathsf{D}_\mathsf{f}(q\|\pi)$ is independent of $q$.[2] Consequently, for any $c > 0$,

$$\pi \in \arg\min_{q\in\mathcal{P}} \mathsf{D}_{\mathrm{KL}}(q\|c\pi).$$

That is, the minimizer is independent of scaling factors on the target density, such as a normalization constant. Reference to a proof of this property of the KL divergence may be found in the bibliography in Section 12.4. ♢

The following result, which we use repeatedly in Chapter 6, is known as the chain rule of KL divergence.

[2] Here we extend the notion of $\mathsf{f}$-divergence in Definition 12.25 to unnormalized densities $\rho$ and $\varrho$.

**Lemma 12.29.** *Let $\rho(u,v)=\rho(u)\rho(v|u)$ and $\varrho(u,v)=\varrho(u)\varrho(v|u)$ be two joint probability densities. It holds that*

$$\mathsf{D}_{\mathrm{KL}}(\rho(u,v)\|\varrho(u,v))=\mathsf{D}_{\mathrm{KL}}(\rho(u)\|\varrho(u))+\mathsf{D}_{\mathrm{KL}}(\rho(v|u)\|\varrho(v|u)),$$

*where*

$$\mathsf{D}_{\mathrm{KL}}(\rho(v|u)\|\varrho(v|u)):=\int\left[\int\log\left(\frac{\rho(v|u)}{\varrho(v|u)}\right)\rho(v|u)\,dv\right]\rho(u)\,du.$$

*Proof.* By direct calculation,

$$\begin{aligned}\mathsf{D}_{\mathrm{KL}}(\rho(v|u)\|\varrho(v|u))&=\int\int\log\left(\frac{\rho(u,v)}{\varrho(u,v)}\right)\rho(u,v)\,dudv\\&=\int\int\log\left(\frac{\rho(u)}{\varrho(u)}\right)\rho(u)\rho(v|u)\,dudv+\int\int\log\left(\frac{\rho(v|u)}{\varrho(v|u)}\right)\rho(u,v)\,dudv\\&=\int\log\left(\frac{\rho(u)}{\varrho(u)}\right)\rho(u)\,du+\int\left[\int\log\left(\frac{\rho(v|u)}{\varrho(v|u)}\right)\rho(v|u)\,dv\right]\rho(u)\,du\\&=\mathsf{D}_{\mathrm{KL}}(\rho(u)\|\varrho(u))+\mathsf{D}_{\mathrm{KL}}(\rho(v|u)\|\varrho(v|u)),\end{aligned}$$

as desired. □

**Example 12.30.** The KL and $\chi^2$ divergences can be explicitly computed for some parametric distributions. For example, if $\rho=\mathcal{N}(\mu_\rho,\Sigma_\rho)$ and $\varrho=\mathcal{N}(\mu_\varrho,\Sigma_\varrho)$ are both $d$-dimensional multivariate Gaussians, the KL and $\chi^2$ divergences are given by

$$\begin{aligned}\mathsf{D}_{\mathrm{KL}}(\rho\|\varrho)&=\frac{1}{2}\left(\mathrm{Tr}(\Sigma_\varrho^{-1}\Sigma_\rho)-d+(\mu_\rho-\mu_\varrho)^\top\Sigma_\varrho^{-1}(\mu_\rho-\mu_\varrho)+\log\left(\frac{\det\Sigma_\varrho}{\det\Sigma_\rho}\right)\right),\\\mathsf{D}_{\chi^2}(\rho\|\varrho)&=\frac{\det(\Sigma_\varrho\Sigma_\rho^{-1})}{\det(2\Sigma_\varrho\Sigma_\rho^{-1}-I)}\exp\Big((\mu_\rho-\mu_\varrho)^\top\Sigma_\rho^{-1}(2\Sigma_\varrho\Sigma_\rho^{-1}-I)^{-1}(\mu_\rho-\mu_\varrho)-1\Big).\end{aligned}\tag{12.14}$$

♢

### 12.2.2 Relationships between f-Divergences and Metrics

Here we establish bounds between some metrics and f-divergences. We continue to work under the simplifying assumption that $\rho,\varrho$ are positive densities.

**Lemma 12.31.** *The Hellinger and total variation metrics are upper bounded by the KL divergence as follows:*

$$\mathsf{D}_{\mathrm{H}}(\rho,\varrho)^2\leq\frac{1}{2}\mathsf{D}_{\mathrm{KL}}(\rho\|\varrho),\quad\mathsf{D}_{\mathrm{TV}}(\rho,\varrho)^2\leq\mathsf{D}_{\mathrm{KL}}(\rho\|\varrho).$$

*Proof.* It suffices to prove only the first inequality since it implies the second by Lemma 12.2. Define function $\varphi:\mathbb{R}^+\mapsto\mathbb{R}$ by

$$\varphi(x):=x-1-\log x.$$

Then

$$\begin{aligned}\varphi'(x) &= 1 - \frac{1}{x},\\ \varphi''(x) &= \frac{1}{x^2},\\ \varphi(\infty) &= \varphi(0) = \infty,\end{aligned}$$

and so the function is convex on its domain. The minimum of $\varphi$ is attained at $x = 1$, and $\varphi(1) = 0$; hence $\varphi(x) \geq 0$ for all $x \in (0, \infty)$. It follows that, for all $x \geq 0$,

$$\begin{aligned}x - 1 &\geq \log x,\\ \sqrt{x} - 1 &\geq \frac{1}{2}\log x.\end{aligned}$$

Using this last inequality we can bound the Hellinger distance as follows:

$$\begin{aligned}\mathsf{D}_{\mathrm{H}}(\rho, \varrho)^2 &= \frac{1}{2}\int \left(1 - \sqrt{\frac{\varrho(u)}{\rho(u)}}\right)^2 \rho(u)\, du\\ &= \frac{1}{2}\int \left(1 + \frac{\varrho(u)}{\rho(u)} - 2\sqrt{\frac{\varrho(u)}{\rho(u)}}\right)\rho(u)\, du\\ &= \int \left(1 - \sqrt{\frac{\varrho(u)}{\rho(u)}}\right)\rho(u)\, du \leq -\frac{1}{2}\int \log\left(\frac{\varrho(u)}{\rho(u)}\right)\rho(u)\, du = \frac{1}{2}\mathsf{D}_{\mathrm{KL}}(\rho\|\varrho).\end{aligned}$$

□

The following lemma shows that the $\chi^2$ divergence upper bounds the KL divergence. By Lemma 12.31 this shows that it also bounds the total variation and Hellinger distances.

**Lemma 12.32.** *The $\chi^2$ divergence upper bounds the KL divergence as follows:*

$$\mathsf{D}_{\mathrm{KL}}(\rho\|\varrho) \leq \log\Big(\mathsf{D}_{\chi^2}(\rho\|\varrho) + 1\Big), \qquad \mathsf{D}_{\mathrm{KL}}(\rho\|\varrho) \leq \mathsf{D}_{\chi^2}(\rho\|\varrho).$$

*Proof.* The second inequality is a direct consequence of the first one, noting that, for $x \geq 0$, $\log(x + 1) \leq x$. To prove the first inequality note that by Jensen inequality

$$\begin{aligned}\mathsf{D}_{\mathrm{KL}}(\rho\|\varrho) &= \int \log\left(\frac{\rho(u)}{\varrho(u)}\right)\rho(u)\, du\\ &\leq \log\left(\int \frac{\rho(u)}{\varrho(u)}\frac{\rho(u)}{\varrho(u)}\varrho(u)\, du\right)\\ &= \log\Big(\mathsf{D}_{\chi^2}(\rho\|\varrho) + 1\Big),\end{aligned}$$

where for the last equality we used that

$$\begin{aligned}
\mathsf{D}_{\chi^2}(\rho\|\varrho) &= \int \left(\frac{\rho(u)}{\varrho(u)} - 1\right)^2 \varrho(u)\,du \\
&= \int \left(\frac{\rho(u)}{\varrho(u)}\right)^2 \varrho(u)\,du - 2\int \left(\frac{\rho(u)}{\varrho(u)}\right) \varrho(u)\,du + \int \varrho(u)\,du \\
&= \int \left(\frac{\rho(u)}{\varrho(u)}\right)^2 \varrho(u)\,du - 1.
\end{aligned}$$

□

### 12.2.3 Invariance of f-Divergences under Invertible Tranformations

This subsection discusses an important property satisfied by all f-divergences: they are invariant under invertible transformations of the underlying variables. Recall the notation for pushforward from the preface. We start with the following lemma relating to pushforwards. The proof is a straightforward consequence of change of variables.

**Lemma 12.33.** *Let $\pi$ be a probability density on $\mathbb{R}^d$ and let $q \in C^1(\mathbb{R}^d, \mathbb{R}^d)$ be invertible everywhere on $\mathbb{R}^d$. Assume further that the determinant of the Jacobian of the inverse,*

$$\mathrm{det}D(q^{-1})(u) = \Bigl(\mathrm{det}Dq\bigl(q^{-1}(u)\bigr)\Bigr)^{-1}, \tag{12.15}$$

*is positive everywhere on $\mathbb{R}^d$. Then, for $u = q(z)$,*

$$\int \varphi(z)\pi(z)\,dz = \int (\varphi \circ q^{-1})(u)(\pi \circ q^{-1})(u)\mathrm{det}D(q^{-1})(u)\,du.$$

*Thus*

$$q_\sharp\pi(u) = (\pi \circ q^{-1})(u)\mathrm{det}D(q^{-1})(u), \tag{12.16a}$$

$$\log q_\sharp\pi(u) = \log(\pi \circ q^{-1})(u) + \log\mathrm{det}D\bigl(q^{-1}(u)\bigr). \tag{12.16b}$$

**Theorem 12.34.** *Let $T\colon \mathbb{R}^d \to \mathbb{R}^d$ be an invertible and differentiable transformation. Then, for two probability density functions $\rho, \varrho$ on $\mathbb{R}^d$ we have*

$$\mathsf{D}_\mathsf{f}(\rho\|\varrho) = \mathsf{D}_\mathsf{f}(T_\sharp\rho\|T_\sharp\varrho).$$

*Proof.* Let $v = T(u)$ denote the transformed variable. For an invertible transformation, $u = T^{-1}(v)$ and $du = \det\bigl(DT^{-1}(v)\bigr)\,dv$. Performing a change of variables in the divergence we have

$$\begin{aligned}
\mathsf{D}_\mathsf{f}(\rho\|\varrho) &= \int \mathsf{f}\left(\frac{\rho(u)}{\varrho(u)}\right)\varrho(u)\,du \\
&= \int \mathsf{f}\left(\frac{\rho\bigl(T^{-1}(v)\bigr)}{\varrho\bigl(T^{-1}(v)\bigr)}\right)\varrho\bigl(T^{-1}(v)\bigr)\det\bigl(DT^{-1}(v)\bigr)\,dv \\
&= \int \mathsf{f}\left(\frac{\rho\bigl(T^{-1}(v)\bigr)\det\bigl(DT^{-1}(v)\bigr)}{\varrho\bigl(T^{-1}(v)\bigr)\det\bigl(DT^{-1}(v)\bigr)}\right)\varrho\bigl(T^{-1}(v)\bigr)\det\bigl(DT^{-1}(v)\bigr)\,dv.
\end{aligned}$$

Using Lemma 12.33 we see that $\rho\big(T^{-1}(v)\big)\det\big(DT^{-1}(v)\big) = T_\sharp\rho(v)$ denotes the density of the pushforward random variable $v$, we have

$$\mathsf{D}_\mathsf{f}(\rho\|\varrho) = \int \mathsf{f}\left(\frac{T_\sharp\rho(v)}{T_\sharp\varrho(v)}\right)T_\sharp\varrho(v)\,dv = \mathsf{D}_\mathsf{f}(T_\sharp\rho\|T_\sharp\varrho).$$

□

## 12.3 Scoring Rules

Probabilistic scoring rules quantify the accuracy of a *forecast distribution* with respect to a *true distribution* known only through samples; the latter is sometimes termed *the verification*. Let $\rho$ and $\varrho$ denote the densities of the forecast and verification distributions, respectively. A *scoring rule* is a function $\mathsf{S}\colon \mathcal{P}(\mathbb{R}^d)\times\mathbb{R}^d \to \mathbb{R}$ that assigns a *score* $\mathsf{S}(\rho, v)$ to a sample $v \sim \varrho$ with respect to the forecast distribution $\rho$. We follow the convention here that scoring rules are negatively oriented meaning that a lower score indicates a better forecast.

To compare the forecast and verification distributions, we define the expected score $\overline{\mathsf{S}}\colon \mathcal{P}(\mathbb{R}^d)\times\mathcal{P}(\mathbb{R}^d)\to\mathbb{R}$ by

$$\overline{\mathsf{S}}(\rho,\varrho) := \mathbb{E}^{v\sim\varrho}[\mathsf{S}(\rho,v)] = \int \mathsf{S}(\rho,v)\varrho(v)\,dv.$$

**Definition 12.35.** A scoring rule $\mathsf{S}$ is called *proper* if, for all $\rho,\varrho\in\mathcal{P}(\mathbb{R}^d)$, $\overline{\mathsf{S}}(\varrho,\varrho)\le\overline{\mathsf{S}}(\rho,\varrho)$. A scoring rule is called *strictly proper* when equality holds if and only if $\rho=\varrho$. ♢

That is, for a proper scoring rule, forecasting with the true distribution results in the lowest expected score. For a strictly proper scoring rule, the lowest expected score can only be attained when forecasting with the true distribution. While a general scoring rule does not need to satisfy the properties in Section 12.2 for statistical distances, it can define a divergence if strict propriety is imposed:

**Lemma 12.36.** *A strictly proper scoring rule can be used to define a divergence between two probability distributions given by*

$$\mathsf{D}_\mathsf{S}(\rho\|\varrho) := \overline{\mathsf{S}}(\rho,\varrho) - \overline{\mathsf{S}}(\varrho,\varrho). \tag{12.17}$$

*Proof.* For a proper scoring rule, the distance is non-negative, i.e., $\mathsf{D}_{\overline{\mathsf{S}}}(\rho,\varrho)\ge 0$. If $\overline{\mathsf{S}}$ is strictly proper, the distance also satisfies the positivity condition, i.e., $\mathsf{D}_{\overline{\mathsf{S}}}(\rho,\varrho)=0$ if and only if $\rho=\varrho$. □

In the following subsections we present five probabilistic scoring rules: the energy score, the continuous ranked probability score, the quantile score, the logarithmic score and the Dawid-Sebastiani score.

### 12.3.1 Energy Score

**Definition 12.37.** Let $\beta \in (0,2)$ be a parameter. For a distribution with probability density function $\rho$ and finite $\beta$-moment, the *energy score* ($\mathsf{ES}$) of a sample $v$ is defined by

$$\mathsf{ES}_\beta(\rho, v) := \mathbb{E}^{u\sim\rho}|u-v|^\beta - \frac{1}{2}\mathbb{E}^{(u,u')\sim\rho\otimes\rho}|u-u'|^\beta. \tag{12.18}$$

♢

We define the expected energy score $\overline{\mathsf{ES}}_\beta \colon \mathcal{P}(\mathbb{R}^d) \times \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}$ by

$$\overline{\mathsf{ES}}_\beta(\rho, \varrho) := \mathbb{E}^{v\sim\varrho}[\mathsf{ES}_\beta(\rho, v)] = \int \mathsf{ES}_\beta(\rho, v)\varrho(v)\, dv. \tag{12.19}$$

**Lemma 12.38.** *Recall the squared energy distance in Definition 12.19, given by*

$$\mathsf{D}^2_{\mathrm{E}}(\rho, \varrho) := 2\mathbb{E}^{(u,v)\sim\rho\otimes\varrho}|u-v| - \mathbb{E}^{(u,u')\sim\rho\otimes\rho}|u-u'| - \mathbb{E}^{(v,v')\sim\varrho\otimes\varrho}|v-v'|. \tag{12.20}$$

*The energy score $\mathsf{ES}_\beta(\rho, v)$ is a strictly proper scoring rule for $\beta \in (0,2)$. Using the distance function in (12.17) with the expected score and with $\beta = 1$, yields*

$$\frac{1}{2}\mathsf{D}^2_{\mathrm{E}}(\rho, \varrho) = \overline{\mathsf{ES}}_1(\rho, \varrho) - \overline{\mathsf{ES}}_1(\varrho, \varrho).$$

*Proof.* It can be shown that the expected energy score (12.19) can be written, for $\beta \in (0,2)$, as

$$\overline{\mathsf{ES}}_\beta(\rho, \varrho) = \frac{\beta 2^{\beta-2}\Gamma(\frac{d}{2}+\frac{\beta}{2})}{\pi^{d/2}\Gamma(1-\frac{\beta}{2})}\int \frac{|\varphi_\rho(u) - \varphi_\varrho(u)|^2}{|u|^{d+\beta}}\, du + \overline{\mathsf{ES}}_\beta(\varrho, \varrho), \tag{12.21}$$

where $\varphi_\rho$ and $\varphi_\varrho$ are the characteristic functions of $\rho$ and $\varrho$, respectively. It follows that the first term is minimized if and only if $\varphi_\rho = \varphi_\varrho$ and hence if and only if $\rho = \varrho$. Since the second term does not depend on $\varrho$, it follows that the energy score is strictly proper for $\beta \in (0,2)$. We give citation for (12.21) in the bibliography; its proof is omitted here for reasons of brevity.

We show the relationship between the energy score and the energy distance as follows:

$$\begin{aligned}
\overline{\mathsf{ES}}_\beta(\rho, \varrho) &= \mathbb{E}^{(u,v)\sim\rho\otimes\varrho}|u-v|^\beta - \frac{1}{2}\mathbb{E}^{(u,u')\sim\rho\otimes\rho}|u-u'|^\beta,\\
\overline{\mathsf{ES}}_\beta(\varrho, \varrho) &= \frac{1}{2}\mathbb{E}^{(u,u')\sim\varrho\otimes\varrho}|u-u'|^\beta.
\end{aligned}$$

Then, for $\beta = 1$,

$$\overline{\mathsf{ES}}_1(\rho, \varrho) - \overline{\mathsf{ES}}_1(\varrho, \varrho) = \mathbb{E}^{(u,v)\sim\rho\otimes\varrho}|u-v| - \frac{1}{2}\mathbb{E}^{(u,u')\sim\rho\otimes\rho}|u-u'| - \frac{1}{2}\mathbb{E}^{(u,u')\sim\varrho\otimes\varrho}|u-u'|,$$

which is indeed half of the squared energy distance in Definition 12.19. □

**Lemma 12.39.** *The energy score* $\mathsf{ES}_2(\rho, v)$ *is proper, but not strictly proper.*

*Proof.* Notice that

$$\begin{aligned}\mathsf{ES}_2(\rho, v) &= \mathbb{E}^{u\sim\rho}|u|^2 - 2\mathbb{E}^{u\sim\rho}[u]^\top v + |v|^2 - \frac{1}{2}(\mathbb{E}^{u\sim\rho}|u|^2 - 2\mathbb{E}^{u\sim\rho}[u]^\top \mathbb{E}^{u\sim\rho}[u] + \mathbb{E}^{u\sim\rho}|u|^2)\\ &= \mathbb{E}^{u\sim\rho}[u]^\top \mathbb{E}^{u\sim\rho}[u] - 2\mathbb{E}^{u\sim\rho}[u]^\top v + |v|^2\\ &= \left|\mathbb{E}^{u\sim\rho}[u] - v\right|^2.\end{aligned}$$

The minimizer of $\mathsf{J}(b) := \mathbb{E}^{v\sim\varrho}|b - v|^2$ is $b^\star = \mathbb{E}^{u\sim\varrho}[u]$, which shows that $\mathsf{ES}_2$ is proper. It is not strictly proper, since for any other distribution $\tilde{\varrho}$ with $\mathbb{E}^{u\sim\varrho}[u] = \mathbb{E}^{u\sim\tilde{\varrho}}[u]$ it holds that $\overline{\mathsf{ES}}_2(\varrho, \varrho) = \overline{\mathsf{ES}}_2(\tilde{\varrho}, \varrho)$. □

Remark 12.40. When $\beta = 2$ the expected energy score is the *mean-square error* of $\mathbb{E}^{u\sim\rho}[u]$. Its square root, the *root-mean-square error (RMSE)* of $\mathbb{E}^{u\sim\rho}[u]$, is often used for evaluating probabilistic forecasts, despite lacking strict propriety. ♢

*Proof of Lemma 12.20.* The proof follows from noting that, from Lemma 12.38 and equation (12.21)

$$\begin{aligned}\mathsf{D}^2_{\mathrm{E}}(\rho, \varrho) &= \overline{\mathsf{ES}}_1(\rho, \varrho) - \overline{\mathsf{ES}}_1(\varrho, \varrho)\\ &= \frac{\Gamma(\frac{d}{2} + \frac{1}{2})}{2\pi^{d/2}\Gamma(\frac{1}{2})} \int \frac{|\varphi_\rho(x) - \varphi_\varrho(u)|^2}{|u|^{d+1}}\, du.\end{aligned}$$

Using the fact that $\mathsf{D}_{\mathrm{E}}(\rho, \varrho)$ is defined as a weighted $L^2-$norm of the difference between the characteristic functions of the pair $(\rho, \varrho)$ leads to the desired metric structure. □

### 12.3.2 Continuous Ranked Probability Score

Let $\mathbb{1}_{u\geq v}$ denote the indicator function of the set $\{u \in \mathbb{R} : u \geq v\}$; this is also known as the Heaviside step function.

**Definition 12.41.** The *continuous ranked probability score* (CRPS) for $\rho \in \mathcal{P}(\mathbb{R})$ and a sample $v \in \mathbb{R}$ is defined by

$$\mathsf{CRPS}(\rho, v) := \int \left(F_\rho(u) - \mathbb{1}_{u\geq v}(u)\right)^2 du. \tag{12.22}$$

♢

We define the expected CRPS $\overline{\mathsf{CRPS}} \colon \mathcal{P}(\mathbb{R}) \times \mathcal{P}(\mathbb{R}) \to \mathbb{R}$ by

$$\overline{\mathsf{CRPS}}(\rho, \varrho) := \mathbb{E}^{v\sim\varrho}[\mathsf{CRPS}(\rho, v)] = \int \mathsf{CRPS}(\rho, v)\varrho(v)\, dv.$$

Note that the indicator function is the cumulative density function of a Dirac mass at $v$ and so the CRPS is comparing two cumulative density functions. The following lemma shows that the above definition of CRPS is equivalent to the energy score in Definition 12.19 with $\beta = 1$.

**Lemma 12.42.**

$$\mathsf{CRPS}(\rho, v) = \mathsf{ES}_1(\rho, v) = \mathbb{E}^{u\sim\rho}|u - v| - \frac{1}{2}\mathbb{E}^{(u,u')\sim\rho\otimes\rho}|u - u'|. \tag{12.23}$$

*Proof.* First, let us recall that the absolute value of a difference can be written as

$$|u - v| = \int_v^\infty \mathbb{1}_{z\leq u}(z)\,dz + \int_{-\infty}^v \mathbb{1}_{z>u}(z)\,dz,$$

by considering the two cases $v \leq u$ and $v > u$. Taking an expectation with respect to $u \sim \rho$ and applying Fubini's theorem we have

$$\begin{aligned}\mathbb{E}^{u\sim\rho}|u - v| &= \int_v^\infty \int_{\mathbb{R}} \mathbb{1}_{z\leq u}(z)\rho(u)\,dudz + \int_{-\infty}^v \int_{\mathbb{R}} \mathbb{1}_{z>u}(z)\rho(u)\,dudz \\ &= \int_v^\infty (1 - F_\rho(z))\,dz + \int_{-\infty}^v F_\rho(z)\,dz \\ &= \int_{\mathbb{R}} (1 - F_\rho(z))\mathbb{1}_{z\geq v}(z)\,dz + \int_{\mathbb{R}} F_\rho(z)\mathbb{1}_{z<v}(z)\,dz. \end{aligned} \tag{12.24}$$

Following similar steps for the second term in (12.23) and taking an expectation over $u' \sim \rho$, we have

$$\frac{1}{2}\mathbb{E}^{(u,u')\sim\rho\otimes\rho}\,|u - u'| = \int_{\mathbb{R}} F_\rho(z)(1 - F_\rho(z))\,dz. \tag{12.25}$$

Lastly, subtracting (12.24) and (12.25) gives

$$\begin{aligned}\mathbb{E}^{u\sim\rho}|u - v| - \frac{1}{2}\mathbb{E}^{(u,u')\sim\rho\otimes\rho}\,|u - u'| &= \int_{\mathbb{R}} \Big(\mathbb{1}_{z\geq v}(z)^2 - 2F_\rho(z)\mathbb{1}_{z\geq v}(z) + F_\rho(z)^2\Big)\,dz \\ &= \int \big(F_\rho(z) - \mathbb{1}_{z\geq v}(z)\big)^2\,dz = \mathsf{CRPS}(\rho, v).\end{aligned}$$

□

The characterization of the CRPS in (12.23) shows that it is composed of two parts: a calibration term that quantifies closeness of the forecast to $v$ and a sharpness term that is related to the spread of the distribution. We note that for a deterministic forecast concentrated on some point $z \in \mathbb{R}$, i.e., $F_\rho(u) = \mathbb{1}_{u\geq z}(u)$, the second term vanishes and the $\mathsf{CRPS}$ reduces to the absolute error between the point $z$ and sample $v$.

**Lemma 12.43.** *The expected CRPS for forecast and verification distributions with probability density functions $\rho$ and $\varrho$, respectively, is given by*

$$\overline{\mathsf{CRPS}}(\rho, \varrho) = \int \big(F_\rho(u) - F_\varrho(u)\big)^2\,du + \int F_\varrho(u)(1 - F_\varrho(u))\,du. \tag{12.26}$$

*Proof.* Start with form (12.22) of the CRPS. Then,

$$\begin{aligned}\mathbb{E}^{v\sim\varrho}\big[\mathsf{CRPS}(\rho, v)\big] &= \int \mathbb{E}^{v\sim\varrho}\Big[\big(F_\rho(u) - \mathbb{1}_{u\geq v}(u)\big)^2\Big]\,du \\ &= \int \mathbb{E}^{v\sim\varrho}\Big[F_\rho(u)^2 - 2F_\rho(u)\mathbb{1}_{u\geq v}(u) + \mathbb{1}_{u\geq v}(u)^2\Big]\,du.\end{aligned}$$

Noting that $\mathbb{1}_{u\geq v}(u)^2 = \mathbb{1}_{u\geq v}(u)$ and that $\mathbb{E}^{v\sim\varrho}[\mathbb{1}_{u\geq v}(u)] = F_\varrho(u)$, we obtain the result after rearranging. □

Remark 12.44. A direct consequence of Lemma 12.43 is that the divergence (12.17) defined by the CRPS takes the form

$$\mathsf{D}_{\mathsf{CRPS}} = \overline{\mathsf{CRPS}}(\rho, \varrho) - \overline{\mathsf{CRPS}}(\varrho, \varrho) = \int \big(F_\rho(u) - F_\varrho(u)\big)^2 \, du,$$

which is known as Cramér's distance. Thus, we have shown that in one dimension CRPS agrees with the energy score $\mathsf{ES}_1$ (Lemma 12.42), the divergence induced by CRPS agrees with Cramer's distance (Lemma 12.43), and the squared energy distance is exactly twice Cramer's distance (Lemma 12.38). ♢

**Example 12.45.** For a univariate Gaussian density $\rho = \mathcal{N}(m, \sigma^2)$, the CRPS with respect to a single observation $v$ has the form

$$\mathsf{CRPS}(\rho, v) = \sigma\left(2\varphi\left(\frac{v-m}{\sigma}\right) + \frac{v-m}{\sigma}\left(2\Phi\left(\frac{v-m}{\sigma}\right) - 1\right) - \frac{2}{\sqrt{\pi}}\right),$$

where $\varphi$ and $\Phi$ denote the probability density function and cumulative density function of a standard Gaussian distribution, respectively. One can observe that the CRPS is minimized for the forecast chosen to have the observation $v$ near the mean $m$ with a small-enough variance to maximize precision. ♢

### 12.3.3 Quantile Score

To motivate the *quantile score* we demonstrate how the *quantile function* can be derived through a minimization problem. To this end let $\alpha \in [0, 1]$. Start by defining the $\alpha-$parameterized family of functions $h_\alpha : \mathbb{R} \to \mathbb{R}$ by

$$h_\alpha(u) = \begin{cases} -(1-\alpha)u, & u < 0, \\ \alpha u, & u \geq 0. \end{cases} \tag{12.27}$$

Function $h_\alpha$ is sometimes termed the *hinge loss*. The hinge loss may also be written

$$h_\alpha(u) = \begin{cases} (1-\alpha)|u|, & u < 0, \\ \alpha|u|, & u \geq 0, \end{cases} \tag{12.28}$$

or as

$$h_\alpha(u) = -(\mathbb{1}_{u \leq 0} - \alpha)u. \tag{12.29}$$

Using this function we define another $\alpha-$parameterized family of functions $L_{\alpha,\varrho} : \mathbb{R} \to \mathbb{R}$ by

$$L_{\alpha,\varrho}(\theta) = \mathbb{E}^{v\sim\varrho}\big[h_\alpha(v - \theta)\big]. \tag{12.30}$$

**Definition 12.46.** Given a probability density function $\varrho$ corresponding to a random variable on $\mathbb{R}$ with cumulative density function $F_\varrho$, the corresponding *quantile function*[3] is $q_\varrho(\alpha) := F_\varrho^{-1}(\alpha)$. ♢

[3] For ease of presentation, we assume throughout invertibility of the cumulative density function; the ideas generalize to the non-invertible case by considering the generalized inverse distribution function.

The quantile function is solution to the $\alpha-$parameterized family of equations $F_\varrho(q_\varrho(\alpha)) = \alpha$. The next lemma shows that this quantile function (of $\alpha$) is also the minimizer of $L_{\alpha,\varrho}$ (viewed as a function of $\alpha$).

**Lemma 12.47.** *For $\alpha \in [0,1]$, it holds that*

$$\operatorname{argmin}_{\theta\in\mathbb{R}} L_{\alpha,\varrho}(\theta) = q_\varrho(\alpha).$$

*Proof.* First note that

$$L_{\alpha,\varrho}(\theta) = (\alpha-1)\int_{-\infty}^{\theta}(v-\theta)\varrho(v)\,dv + \alpha\int_{\theta}^{\infty}(v-\theta)\varrho(v)\,dv.$$

Differentiating with respect to $\theta$ yields

$$\begin{aligned}\frac{d}{d\theta}L_{\alpha,\varrho}(\theta) &= (1-\alpha)\int_{-\infty}^{\theta}\varrho(v)\,dv - \alpha\int_{\theta}^{\infty}\varrho(v)\,dv\\ &= \int_{-\infty}^{\theta}\varrho(v)\,dv - \alpha\\ &= F_\varrho(\theta) - \alpha.\end{aligned}$$

Setting the derivative to zero yields the desired result. □

**Definition 12.48.** For $\rho \in \mathcal{P}(\mathbb{R})$, the *quantile score* at level $\alpha$, verified against $v \sim \varrho$, is defined as

$$\mathsf{QS}_\alpha(\rho, v) \coloneqq \Big(\mathbb{1}_{v\le q_\rho(\alpha)} - \alpha\Big)(q_\rho(\alpha) - v). \tag{12.31}$$

♢

We define the expected quantile score $\overline{\mathsf{QS}}_\alpha\colon \mathcal{P}(\mathbb{R})\times\mathcal{P}(\mathbb{R}) \to \mathbb{R}$ by

$$\overline{\mathsf{QS}}_\alpha(\rho,\varrho) \coloneqq \mathbb{E}^{v\sim\varrho}[\mathsf{QS}_\alpha(\rho,v)] = \int \mathsf{QS}_\alpha(\rho,v)\varrho(v)\,dv.$$

The quantile score arises from evaluating the hinge loss in (12.29) at $u = v - F_\rho^{-1}(\alpha)$, i.e. $\mathsf{QS}_\alpha(\rho,v) = h_\alpha(v - F_\rho^{-1}(\alpha))$. Note that constant function taking value $v$ in $(0,1)$ is the quantile function associated with the Dirac mass at $v$: for $\alpha \in (0,1)$, we have that $F_{\delta_v}^{-1}(\alpha) = \inf\{u : F_{\delta_v}(u) \ge \alpha\} = v$, where here we have used the generalized inverse distribution. Thus equation (12.31) is comparing the value of two quantile functions evaluated at $\alpha$. The following is a direct consequence of Lemma 12.47:

**Proposition 12.49.** *Define*

$$\rho^\star = \operatorname{argmin}_\rho \overline{\mathsf{QS}}_\alpha(\rho,\varrho).$$

*Then, for all $\alpha \in [0,1]$, $\rho^\star = \varrho$ is a solution of the minimization problem.*

*Proof.* Note that

$$\begin{aligned}\mathbb{E}^{v\sim\varrho}[\mathsf{QS}_\alpha(\rho,v)] &= \mathbb{E}^{v\sim\varrho}\Big[h_\alpha(v - F_\rho^{-1}(\alpha))\Big]\\ &= L_{\alpha,\varrho}(q_\rho(\alpha)).\end{aligned}$$

By Lemma 12.47 the optimal $\rho$ will be one for which $q_\rho(\alpha) = q_\varrho(\alpha)$. This can be achieved by setting $\rho = \varrho$. □

The following result shows the CRPS can also be computed as twice the integral of the quantile score over all quantiles.

**Lemma 12.50.**

$$\mathsf{CRPS}(\rho, v) = 2\int_0^1 \mathsf{QS}_\alpha(\rho, v)\,d\alpha.$$

*Hence*

$$\overline{\mathsf{CRPS}}(\rho, \varrho) = 2\int_0^1 \overline{\mathsf{QS}}_\alpha(\rho, \varrho)\,d\alpha.$$

*Proof.* Applying the change of variables $u = F_\rho^{-1}(\alpha)$ to the integrated quantile score with $d\alpha = \rho(u)\,du$, we have

$$\begin{aligned} 2\int_0^1 \mathsf{QS}_\alpha(\rho, v)\,d\alpha &= \int_0^1 2\Big(\mathbb{1}_{v \le F_\rho^{-1}(\alpha)} - \alpha\Big)\Big(F_\rho^{-1}(\alpha) - v\Big)\,d\alpha \\ &= 2\int_{\mathbb{R}} (\mathbb{1}_{v\le u} - F_\rho(u))(u-v)\rho(u)\,du. \end{aligned}$$

We recognize that, for $u \neq v$, $\frac{d}{du}\big(\mathbb{1}_{v\le u} - F_\rho(u)\big)^2 = -2(\mathbb{1}_{v\le u} - F_\rho(u))\rho(u)$. Applying integration by parts gives us

$$\begin{aligned} 2\int_0^1 \mathsf{QS}_\alpha(\rho, v)\,d\alpha &= -\int_{\mathbb{R}} \frac{d}{du}(\mathbb{1}_{v\le u} - F_\rho(u))^2(u-v)\,du \\ &= \big(\mathbb{1}_{v\le u} - F_\rho(u)\big)^2(u-v)\Big|_{u=-\infty}^{u=\infty} + \int_{\mathbb{R}} \big(\mathbb{1}_{v\le u} - F_\rho(u)\big)^2\,du. \end{aligned}$$

Now we claim that the boundary terms in the integration by parts vanish. Indeed, by Markov's inequality it holds that $1 - F_\rho(u) \le \frac{\mathbb{E}^{w\sim\rho}|w|}{u}$ and so $\big(1 - F_\rho(u)\big)^2 u \to 0$ as $u \to \infty$. The limit as $u \to -\infty$ is similar. Hence, we have shown that

$$2\int_0^1 \mathsf{QS}_\alpha(\rho, v)\,d\alpha = \int_{\mathbb{R}} \big(\mathbb{1}_{v\le u} - F_\rho(u)\big)^2\,du = \mathsf{CRPS}(\rho, v),$$

as desired. □

### 12.3.4 Logarithmic Score

The logarithmic score evaluates the negative logarithm of the forecast probability density function at a sample $v \sim \varrho$ :

**Definition 12.51.** The *logarithmic score* is

$$\mathsf{LS}(\rho, v) := -\log \rho(v). \tag{12.32}$$

♢

We define the expected logarithmic score $\overline{\mathsf{LS}}\colon \mathcal{P}(\mathbb{R}^d) \times \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}$ by

$$\overline{\mathsf{LS}}(\rho, \varrho) := \mathbb{E}^{v\sim\varrho}[\mathsf{LS}(\rho, v)] = \int \mathsf{LS}(\rho, v)\varrho(v)\,dv.$$

The logarithmic score is also sometimes referred to as the *ignorance* when the logarithm is in base 2. Notice that the logarithmic score penalizes heavily forecasts $\rho$ that place low probability in outcomes that materialize: for small $\rho(v)$, $\mathsf{LS}(\rho, v) := -\log \rho(v)$ is very large.

**Lemma 12.52.** *The logarithmic score is a strictly proper scoring rule. The divergence resulting from the expected logarithmic score is the KL divergence in Definition 12.27 between $\varrho$ and $\rho$:*

$$\mathsf{D}_{\mathrm{KL}}(\varrho\|\rho) = \overline{\mathsf{LS}}(\rho, \varrho) - \overline{\mathsf{LS}}(\varrho, \varrho). \tag{12.33}$$

*Proof.* We have that

$$\overline{\mathsf{LS}}(\rho, \varrho) = \mathbb{E}^{v\sim\varrho}[\mathsf{LS}(\rho, v)] = -\int \log \rho(v)\varrho(v)\, dv,$$
$$\overline{\mathsf{LS}}(\varrho, \varrho) = \mathbb{E}^{v\sim\varrho}[\mathsf{LS}(\varrho, v)] = -\int \log \varrho(v)\varrho(v)\, dv.$$

Recalling that

$$\mathsf{D}_{\mathrm{KL}}(\varrho\|\rho) = \int \log\left(\frac{\varrho(v)}{\rho(v)}\right)\varrho(v)\, dv$$

delivers the desired result. □

### 12.3.5 Dawid–Sebastiani Score

The Dawid–Sebasatiani score controls the first and second moments of the forecast:

**Definition 12.53.** The *Dawid–Sebastiani* score is

$$\mathsf{DS}(\rho, v) := |v - m|_C^2 + \log(\det(C)), \tag{12.34}$$

where $m$ and $C$ are the mean and covariance of $\rho$. ♢

It is easy to verify that the Dawid–Sebastiani score is equivalent (up to a linear transformation) to the logarithmic score in the Gaussian case, but can be used for general distributions with finite mean and covariance. The Dawid–Sebastiani score is proper, but not strictly proper. This is similar to the $\mathsf{ES}_2$ score, which reduces to controlling the first moment; the Dawid–Sebastiani score controls first and second moments.

### 12.3.6 Noise in the Verification

In the presence of noise in the verification, the scoring rule will generally favor more dispersed forecasts than if the noise were not present. To account for this noise, either the forecast distribution or the scoring rule itself can be modified using knowledge about the noise distribution.

Suppose that the true value of the predictand is $v \sim \varrho$, but that the verification is given by $\widetilde{v}|v \sim r$, where $r(\widetilde{v}|v)$ is the conditional noise distribution. We can then modify the definition of propriety 12.35 as follows:

**Definition 12.54.** A scoring rule $\mathsf{S}$ is called proper with respect to the noise distribution $r$ if for all $\rho$, $\mathbb{E}^{\widetilde{v}\sim\widetilde{\varrho}}[\mathsf{S}(\varrho, \widetilde{v})] \le \mathbb{E}^{\widetilde{v}\sim\widetilde{\varrho}}[\mathsf{S}(\rho, \widetilde{v})]$, where $\widetilde{\varrho}(\widetilde{v}) = \int r(\widetilde{v}|v)\varrho(v)\, dv$. It is called strictly proper when equality holds if and only if $\rho = \varrho$. ♢

**Proposition 12.55.** *If* $\mathsf{S}$ *is proper, then* $\widetilde{\mathsf{S}}(\rho, \widetilde{v}) := \mathsf{S}(\widetilde{\rho}, \widetilde{v})$ *is proper with respect to* $r$*, where*

$$\widetilde{\rho}(\widetilde{v}) := \int r(\widetilde{v}|v)\rho(v)\, dv.$$

*Proof.* We have

$$\begin{aligned}\mathbb{E}^{\widetilde{v}\sim\widetilde{\varrho}}[\widetilde{\mathsf{S}}(\rho,\widetilde{v})] &= \mathbb{E}^{\widetilde{v}\sim\widetilde{\varrho}}[\mathsf{S}(\widetilde{\rho},\widetilde{v})]\\ &\geq \mathbb{E}^{\widetilde{v}\sim\widetilde{\varrho}}[\mathsf{S}(\widetilde{\varrho},\widetilde{v})]\\ &= \mathbb{E}^{\widetilde{v}\sim\widetilde{\varrho}}[\widetilde{\mathsf{S}}(\varrho,\widetilde{v})],\end{aligned}$$

where the second line follows from the propriety of $\mathsf{S}$. □

Note that propriety does not guarantee an accurate ranking of forecast distributions if the forecast distributions are evaluated on verifications with different levels of noise. To ensure this property, the scoring rule must be modified for each level of noise such that the expected score is the same as it would be if there were no noise. A scoring rule possessing this property is known as *unbiased*.

### 12.3.7 Distance-Like Deterministic Scoring Rule

**Definition 12.56.** We call a function $\mathsf{D}(\cdot,\cdot):\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ a *distance-like deterministic scoring rule* if it satisfies the following three properties for all $u,v\in\mathbb{R}^d$ :

1. Non-negative: $\mathsf{D}(u,v)\geq 0$.
2. Positive: $\mathsf{D}(u,v)=0$ if and only if $u=v$.
3. Symmetric: $\mathsf{D}(u,v)=\mathsf{D}(v,u)$.

♢

**Example 12.57.** Let $\mathsf{D}(u,v)=\|u-v\|^p$ for any norm $\|\cdot\|$ on $\mathbb{R}^d$ and any $p\in(0,\infty)$. Then $\mathsf{D}(\cdot,\cdot)$ is a distance-like scoring rule. ♢

## 12.4 Bibliography

We recommend the paper [113] for study of metrics, and other distance-like functions, including divergences, on the set of probability measures. The paper [58] contains proof of the assertion made in Remark 12.28. The energy distance is overviewed in [304]. Relationships between energy distance and MMD may be found in [287]. Relating the characteristic property (and universal property of positive definite kernels) to mean embedding of measures is the subject of [296]. An overview of probabilistic scoring rules is given in [114]. The CRPS was first introduced in [43] and [213]. The relationship between the quantile score and the CRPS stated in Lemma 12.50 can be found in [177]; see also [92, 115]. The discussion of bias in the ensemble CRPS is given in [100]. Closed-form expressions for the CRPS for various distributions are given in [164], and references to some additional known expressions are given in Table 9.7 of [333]. Although not employed here, we mention the Fréchet inception distance from [140] which is widely used to evaluate image generation.

Reproducing kernel Hilbert spaces have become a useful framework to understand and develop theoretical results for machine learning methods. The background and properties of RKHSs can be found in [27, 290]; see also [224] for a broad survey on kernel mean embeddings. Some applications of RKHSs in machine learning include comparing distributions based on two-sample tests from estimators for the maximum-mean discrepancy [121] and measuring (conditional) dependence between random variables [120]. The statistical properties of maximum mean discrepancy have also made it a popular metric for generative modeling [108].

A framework for scoring rules in the presence of observation error in the verification is given in [95]. An extensive discussion of scoring rules, with focus on meteorological applications, is given in Chapter 9 of [333]. The relationship between the CRPS and the spread–error relationship is discussed in [193]. The identity (12.21) is proved by using Proposition 2 in [304]. Another quantity that is widely used in the verification of spatial fields, particularly in weather forecasting, is the anomaly correlation coefficient [226, 333].

# Chapter 13

## Unsupervised Learning and Generative Modeling

In the field of *unsupervised learning* the following data assumption is made:

**Data Assumption 13.1.** *We have available data in the form*

$$U := \{u^{(n)}\}_{n=1}^{N}, \tag{13.1}$$

*assumed to be drawn i.i.d. from probability density function* $\Upsilon$ *on* $\mathbb{R}^d$ *which is unknown.*

Through the data, we have access to the empirical density

$$\Upsilon^{\mathrm{N}}(u) = \frac{1}{N}\sum_{n=1}^{N} \delta(u - u^{(n)}). \tag{13.2}$$

The goal is to understand the data set summarized by measure $\Upsilon^{\mathrm{N}}$. One widely used approach is to study *clustering* within the data; we will not pursue this important topic because it is not of direct relevance to the solution of inverse problems and data assimilation, our focus in these notes. Instead we pursue the approach of *generative modeling*: we try and create new samples from $\Upsilon$, given $\Upsilon^{\mathrm{N}}$. We illustrate the idea of generative modeling with two examples.

**Example 13.2.** A natural starting point for generative modeling is to discuss *density estimation*. This methodology seeks to approximate $\Upsilon(u)$, from within some class of densities, on the basis of the data summarized in $\Upsilon^{\mathrm{N}}(u)$. The goal is not always generative modeling: it may be to have a smooth density which can be differentiated or otherwise manipulated in a way that $\Upsilon^{\mathrm{N}}(u)$ cannot. However, some classes of approximate $\Upsilon$ can be used to generate new samples. An example is when the approximation is a Gaussian mixture. ♢

**Example 13.3.** We introduce the *measure transport* task of determining function $g : \mathbb{R}^{d_z} \to \mathbb{R}^d$ so that, given probability density function $\zeta$ on $\mathbb{R}^{d_z}$ and probability density function $\Upsilon$ on $\mathbb{R}^d$, $g_\sharp\zeta = \Upsilon$; here $g_\sharp$ denotes the pushforward operation[1] and so, explicitly, we are seeking $g$ so that, if $z \sim \zeta$, then $g(z) \sim \Upsilon$. The task of identifying $g$ must be undertaken empirically, since only $\Upsilon^{\mathrm{N}}$ is available to us, not $\Upsilon$. A common setting is

[1] Recall that pushforward is defined in the preface.

to assume that $\zeta$ is Gaussian. Then the aim is to find map $g$ so that samples from Gaussian $\zeta$, when pushed forward under $g$, will look like samples from $\Upsilon$. We will also study relaxations of this problem, in which the Gaussian $\zeta$ is convolved with a Gaussian whose mean is function $g$. ♢

In Section 13.1 we describe the subject of density estimation, introduced in Example 13.2, and taking a particular perspective on it which links to several themes arising in other places in the notes. In the remainder of the chapter we study variants on the idea illustrated in Example 13.3. All the methods connect the target measure $\Upsilon$, which we wish to sample, and defined on $\mathbb{R}^d$, to a measure on a latent space $\mathbb{R}^{d_z}$. In Section 13.2 we study a general approach to measure transport akin to the density estimation approach studied in Section 13.1; typically these methods employ $d_z = d$. Normalizing flows are studied in 13.3; whilst $d_z \neq d$ is possible here too, in the continuum limit, using neural ODEs, this methodology has at its core an invertible mapping with $d_z = d$. In Section 13.4 we study score-based approaches that build on Langevin sampling. Sections 13.5 and 13.6 introduce autoencoders and variational autoencoders respectively, where an approximate inverse for $g$ is sought, although typically $d_z < d$ so caution is required to define this carefully. Generative adversarial networks (GANs) are introduced in Section 13.7, applicable with $d_z \neq d$.

## 13.1 Density Estimation

Consider the problem of estimating the probability density function underlying data $U := \{u^{(n)}\}_{n=1}^N$; we assume the data points $u^{(n)} \in \mathbb{R}^d$ are drawn i.i.d. from the same distribution. Let $\mathcal{P}(\mathbb{R}^d)$ denote the set of all probability density functions on $\mathbb{R}^d$. We seek to solve the problem by finding a probability density function from a tractable class of probability distributions $\mathcal{Q} \subset \mathcal{P}(\mathbb{R}^d)$, for the purpose of computations and for the purpose of revealing an explicit, interpretable form. To be concrete we assume that, for $\Theta \subseteq \mathbb{R}^p$, $\mathcal{Q}$ comprises a set of probability density functions $q(\cdot;\theta) \in \mathcal{P}(\mathbb{R}^d)$ for each $\theta \in \Theta$.

Our objective is to find $\mathbb{P}(\theta|U)$ by applying Bayesian inference, as in Chapter 1. If we place prior $\mathbb{P}(\theta)$ on unknown parameter $\theta$, then, noticing that $\mathbb{P}(U|\theta) = \Pi_{n=1}^N q(u^{(n)};\theta)$, we obtain from Theorem 1.2 that

$$\mathbb{P}(\theta|U) \propto \Pi_{n=1}^N q(u^{(n)};\theta)\mathbb{P}(\theta). \tag{13.3}$$

If we seek a maximum a posteriori (MAP) estimator for $\theta$, then, following the developments in Section 1.2,

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{J}^N_{\text{MAP}}(\theta),$$
$$\mathsf{J}^N_{\text{MAP}}(\theta) = -\sum_{n=1}^N \log q(u^{(n)};\theta) - \log \mathbb{P}(\theta).$$

Indeed in the nomenclature of Section 1.2 the first term in the definition of $\mathsf{J}^N_{\text{MAP}}(\cdot)$ is the loss function (1.5) and the second is the regularizer (1.6). If we drop the regularizer, corresponding to what is sometimes termed a *flat prior*, then we obtain the maximum likelihood estimation (MLE) problem

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{J}^N_{\text{MLE}}(\theta),$$
$$\mathsf{J}^N_{\text{MLE}}(\theta) = -\sum_{n=1}^{N} \log q(u^{(n)};\theta).$$

This MLE may be derived by an alternative methodology. Consider the minimization problem

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{F}(\theta),$$
$$\mathsf{F}(\theta) = \mathsf{D}_{\text{KL}}\big(\Upsilon \| q(\cdot;\theta)\big).$$

Notice that the KL divergence defining the objective $\mathsf{F}$ is given by

$$\mathsf{D}_{\text{KL}}\big(\Upsilon \| q(\cdot;\theta)\big) = \mathbb{E}^{u\sim\Upsilon}\Big[\log \Upsilon(u) - \log q(u;\theta)\Big],$$

which involves the unknown density $\Upsilon$. However, since optimization is over the parameter $\theta$ in the density $q(\cdot;\theta)$, minimizing $\mathsf{F}$ over $\theta$ is the same as minimizing

$$-\mathbb{E}^{u\sim\Upsilon}\Big[\log q(u;\theta)\Big]$$

over $\theta$. Replacing expectation with respect to $\Upsilon$ with expectation with respect to the empirical density $\Upsilon^{\text{N}}$ defined in (13.2), we obtain

$$-\mathbb{E}^{u\sim\Upsilon^{\text{N}}}\Big[\log q(u;\theta)\Big] = -\frac{1}{N}\sum_{n=1}^{N} \log q(u^{(n)};\theta) = \frac{1}{N}\mathsf{J}^N_{\text{MLE}}(\theta).$$

## 13.2 Transport Methods

Let $\Upsilon, \zeta \in \mathcal{P}(\mathbb{R}^d)$. We now consider the task of finding invertible $g : \mathbb{R}^d \to \mathbb{R}^d$ with the property that $\Upsilon \approx g_\sharp\zeta$; equivalently, since $g$ is invertible, $\zeta \approx (g^{-1})_\sharp\Upsilon$. We seek to determine a parameter such that $g$ chosen from the parametric class $g : \mathbb{R}^d \times \Theta \to \mathbb{R}^d$, with $\Theta \subseteq \mathbb{R}^p$, realizes this approximation. To this end we consider the optimization problem

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{F}(\theta),$$
$$\mathsf{F}(\theta) = \mathsf{D}_{\text{KL}}\big(\Upsilon \| g(\cdot;\theta)_\sharp\zeta\big).$$

We define the resulting approximate pushforward map by $g^\star = g(\cdot;\theta^\star)$. Notice that this approach agrees with the density estimation approach in Section 13.1, now parameterizing the approximating densities $q(\cdot;\theta) = g(\cdot;\theta)_\sharp\zeta$ as pushfoward of a fixed reference density $\zeta$ by a parameterized transport map $g(\cdot;\theta)$.

Remark 13.4. Note that there exist perfect transport maps $g_{\text{perfect}}$, for example an optimal transport, for which $\mathsf{D}_{\text{KL}}(\Upsilon \| (g_{\text{perfect}})_\sharp \zeta) = 0$. We aim to get as close as possible to a perfect transport solution, within our parametric class, by minimizing $\mathsf{F}$ over $\theta$. In practice, when $\Upsilon$ is unknown this task needs to be undertaken based on the data summarized in the empirical measure $\Upsilon^{\mathrm{N}}$. Whilst optimal transport —studied in Section 12.1.3 in the context of transport distances between probability distributions— provides a deep mathematical structure within which to consider the measure transport problem, there are numerous applications where the optimality constraint on the transport confers few advantages over other transports, and may unnecessarily complicate the computational task of finding the transport. For these reasons, in this chapter we focus on other approaches to transport that relax the optimality constraint, but preserve the core idea of sampling probability distributions by constructing transports from a known distribution (that we know how to sample) to the target, and in particular they resonate with the Monge formulation of optimal transport. ♢

As in Section 13.1, to optimize $\mathsf{F}(\cdot)$ we will identify $\mathsf{J}(\cdot)$, a function which differs from the KL divergence only by a constant independent of $\theta$, and optimize $\mathsf{J}(\cdot)$. Lemma 12.33 is useful in determining an explicit form of $\mathsf{J}$ for computational purposes. We now use this lemma to find a pair of expressions for $\mathsf{J}(\cdot)$ that are useful for a variety of specific instances of the general measure transport problem of interest. Using the expression for the KL divergence we find that

$$\mathsf{F}(\theta) = \mathsf{D}_{\text{KL}}(\Upsilon \| g(\cdot;\theta)_\sharp \zeta) = \mathbb{E}^{u\sim\Upsilon}\Big[\log \Upsilon(u) - \log g_\sharp \zeta(u)\Big].$$

Thus, noting that the desired minimization is independent of $\theta-$independent constants, and using (12.16b) we define

$$\mathsf{J}(\theta) = -\mathbb{E}^{u\sim\Upsilon}\Big[\log \zeta \circ g^{-1}(u;\theta) + \log \det D_u (g^{-1})(u;\theta)\Big]. \tag{13.4}$$

Note that $\mathsf{F}(\theta) = \mathsf{J}(\theta) + c$ where constant $c$ is independent of parameter $\theta$ to be optimized over. Our desired minimization problem for $\mathsf{F}(\cdot)$ is thus equivalent to minimizing (13.4). Furthermore, using (12.15) in (13.4), we may write

$$\mathsf{J}(\theta) = -\mathbb{E}^{u\sim\Upsilon}\Big[\log \zeta \circ g^{-1}(u;\theta) - \log \det D_u g(g^{-1}(u;\theta);\theta)\Big]. \tag{13.5}$$

Remark 13.5. In practice, if $\Upsilon$ is only available empirically, objective (13.4) or (13.5) can be approximated as follows:

$$\begin{aligned}\mathsf{J}^N(\theta) &= -\frac{1}{N}\sum_{n=1}^{N}\Big[\log \zeta \circ g^{-1}(u^{(n)};\theta) + \log \det D_u (g^{-1})(u^{(n)};\theta)\Big] \\ &= -\frac{1}{N}\sum_{n=1}^{N}\Big[\log \zeta \circ g^{-1}(u^{(n)};\theta) - \log \det D_u g(g^{-1}(u^{(n)};\theta);\theta)\Big].\end{aligned} \tag{13.6}$$

Then the forward map $g^\star$ is obtained at $\theta^\star$ which minimizes $\mathsf{J}^N(\cdot)$. By minimizing (13.6) we thus find an approximate expression for $\Upsilon$, known only through samples, as the pushforward under $g^\star$ of $\zeta$. Once we have the map $g^\star$ we can generate new (approximate) samples from $\Upsilon$ by sampling $\zeta$ and applying $g^\star$. An important point to note here is that, to determine $g^\star$, minimization of either form of the loss function $\mathsf{J}^N$ in (13.6) requires evaluation of $g^{-1}$. Thus $g$ must be readily invertible. Section 13.3 addresses this issue in the context of normalizing flows. $\diamondsuit$

Remark 13.6. The ideas in this section are also useful in the context where $\Upsilon$ is a known simple measure, from which samples are easily drawn, and $\zeta$ is a more complicated measure which we wish to characterize and sample from; in such a setting, which we consider in Section 5.1, the objective is to determine $T = g^{-1}$ so that $\zeta = (g^{-1})_\sharp \Upsilon$. Since (13.4) is expressed entirely in terms of $T = g^{-1}$, and not $g$ itself, it is possible to approach this problem by directly parameterizing $T = g^{-1}$ rather than $g$. $\diamondsuit$

## 13.3 Normalizing Flows

The transport approach from Section 13.2 requires invertibility of the map $g$, and efficient computation of the determinant of the Jacobian. Invertibility of the map $g$ is also useful in other contexts. Normalizing flows address this issue; *normalizing* reflects the fact that distribution $\zeta$ is often a Gaussian, whilst *flow* connotes the breaking up of the map $g$ into a sequence of simpler maps which are themselves parameterized, rather than parameterizing $g$ itself.

To this end we fix $J \in \mathbb{N}$ and introduce the iteration

$$v_{j+1} = H(v_j;\theta), \quad j = 0, \dots, J-1, \tag{13.7a}$$

$$v_0 = z, \tag{13.7b}$$

where we define $u := v_J$. Then $u = g(z;\theta)$ where

$$g(\cdot;\theta) = H(\cdot;\theta) \circ H(\cdot;\theta) \circ \cdots \circ H(\cdot;\theta), \tag{13.8}$$

the $J-$fold composition of $H(\cdot;\theta)$. The proposed method is thus simply a transport of the type discussed in Section 13.2, with a specific construction of $g$ in compositional form. The remainder of the section comprises two components: firstly showing specifics of the formulation of the minimization problems (13.4) or (13.5) in this compositional setting; and secondly showing a continuum limit of the compositional setting, which recovers neural ODEs.

### 13.3.1 Structure in the Optimization Problem for $\theta$

We assume that $H(\cdot;\theta)$ is invertible, with inverse $H^{-1}(\cdot;\theta)$, and differentiable, with derivative $DH(\cdot;\theta)$. Assume that $z \sim \zeta$ and let $p_j(v_j)$ denote the density of $v_j$. Then the goal of normalizing flow is to choose parameter $\theta$ in $H$ so that $p_J \approx \Upsilon$ and hence (approximately) $u \sim \Upsilon$; this is potentially a more straightforward task than working with a directly parameterized $g$, the approach described in Section 13.2.

Formulae (12.15), (12.16) show that, since $v_j = H^{-1}(v_{j+1};\theta)$,

$$\log p_{j+1}(v_{j+1}) = \log p_j(v_j) - \log \det DH(v_j;\theta).$$

By induction, we obtain that

$$\log p_J(v_J) = \log p_0(v_0) - \sum_{j=0}^{J-1} \log \det DH(v_j;\theta),$$

and hence

$$\log g_\sharp \zeta(u) = \log \zeta(z) - \sum_{j=0}^{J-1} \log \det DH(v_j;\theta), \tag{13.9}$$

where $u = g(z;\theta)$ and $v_j = H^{(j)}(z;\theta)$, which we use to denote the $j-$fold composition of $H(\cdot;\theta)$. Note that $v_j = (H^{-1})^{(J-j)}(u;\theta)$, the $(J-j)-$fold composition of $H^{-1}(\cdot;\theta)$. In particular $z$ is found as the $J-$fold composition of $H^{-1}(\cdot;\theta)$ evaluated at $u$.

With these relationships defined between $(z,\{v_j\})$ and $u$, and noting that they are parameterized by $\theta$, we obtain from the general loss function (13.5)

$$\mathsf{J}(\theta) = -\mathbb{E}^{u\sim\Upsilon}\Big[\log \zeta(z) - \sum_{j=0}^{J-1} \log \det DH(v_j;\theta)\Big]. \tag{13.10}$$

As before we set

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{J}(\theta)$$

and define the resulting approximate pushforward map by $g^\star = g(\cdot;\theta^\star)$. Note that $g(\cdot;\theta^\star)$ is found from the $J-$fold composition of $H(\cdot;\theta^\star)$. Again, in practice the expectation over $\Upsilon$ in (13.10) can be approximated using the empirical density $\Upsilon^{\mathrm{N}}$.

Remark 13.7. The preceding setting can be generalized to one in which each map in the composition is different and carries its own set of parameters to be optimized, so that (13.8) is replaced by

$$g(\cdot;\theta) = H_{J-1}(\cdot;\theta^{(J-1)}) \circ H_{J-2}(\cdot;\theta^{(J-2)}) \circ \cdots \circ H_0(\cdot;\theta^{(0)}), \tag{13.11}$$

and $\theta = \{\theta^{(j)}\}_{j=0}^{J-1}$. ♢

### 13.3.2 Neural ODEs

Neural ODEs map a probability distribution $\zeta$ on the initial condition into a probability distribution $g_\sharp\zeta$ on the solution of an ODE at time $t=1$; here $g$ is the solution map of an ODE with flow defined by a parameterized vector field.

A continuum limit of normalizing flows can be used to construct neural ODEs. To this end we define

$$\begin{aligned} \frac{dv}{dt} &= h(v;\theta), \\ v(0) &= z. \end{aligned} \tag{13.12}$$

Noting that $v(t)$ depends on $z$ we set $g_t(z;\theta) = v(t)$ and $g(z;\theta) = g_1(z;\theta) = v(1)$. An advantage of the continuum perspective is that the inverse of map $g$ is readily computed as follows. Define the equation

$$\begin{aligned}\frac{dw}{dt} &= -h(w;\theta),\\ w(0) &= u.\end{aligned}$$

Then $g_t^{-1}(u;\theta) = w(t)$ and, in particular, $g^{-1}(u;\theta) = w(1)$.

**Lemma 13.8.** *Assume that in* (13.12) *the initial condition satisfies $z \sim \zeta$. Then*

$$\log g_\sharp\zeta(u) = \log\zeta(z) - \int_0^1 \operatorname{div} h\big(v(s);\theta\big)\, ds. \tag{13.13}$$

*Proof.* This may be derived from the discrete normalizing flow, by setting $H(\cdot;\theta) = \mathrm{Id} + \Delta t h(\cdot;\theta)$ where $J\Delta t = 1$. Note that this corresponds to an Euler approximation of the continuum picture in which $v_j \approx v(j\Delta t)$. Furthermore

$$\det DH = \det(I_d + \Delta t Dh) = 1 + \Delta t \mathrm{Tr} Dh + \mathcal{O}(\Delta t^2).$$

Now note that $\mathrm{Tr}\, Dh = \operatorname{div} h$. Thus, substituting this expression for $\det DH$ into (13.9), expanding the logarithm in powers of $\Delta t$, summing over $j$ and letting $\Delta t \to 0$ yields the desired result. □

The continuum analog of (13.10) is then

$$\mathsf{J}(\theta) = -\mathbb{E}^{u\sim\Upsilon}\Big[\log\zeta(z) - \int_0^1 \operatorname{div} h\big(v(s);\theta\big)\, ds\Big], \tag{13.14}$$

where $z = g^{-1}(u;\theta)$ and $v(t) = g_{1-t}^{-1}(u;\theta)$. As before, up to an additive constant, $\mathsf{J}(\theta) = \mathsf{D}_{\mathrm{KL}}(\Upsilon \| g_\sharp\zeta)$. Minimizing $\mathsf{J}$ over $\theta$, to obtain $\theta^\star$, thus leads to $g^\star(\cdot) = g(\cdot;\theta^\star)$ and $(g^\star)_\sharp\zeta \approx \Upsilon$.

Remark 13.9. Using the setting of Remark 13.7 leads to a generalized form of neural ODE taking the form

$$\begin{aligned}\frac{dv}{dt} &= h\big(v,t;\theta(t)\big),\\ v(0) &= z.\end{aligned} \tag{13.15}$$

Optimization is now over a function of time, $\theta(t)$, and some form of regularization is needed to enforce continuity of $\theta(\cdot)$. For example we may replace (13.14) by

$$\mathsf{J}(\theta) = -\mathbb{E}^{u\sim\Upsilon}\Big[\log\zeta(z) - \int_0^1 \operatorname{div} h\big(v(s),s;\theta(s)\big)\, ds\Big] + \lambda\int_0^1 \Big|\frac{d\theta}{dt}(s)\Big|^2 ds. \tag{13.16}$$

Notice that $z = v_0$ and $v(s)$ are functions of $u$, defined by the backward evolution of the ODE (13.15). ♢

## 13.4 Score-Based Approaches

We have overviewed a variety of techniques for generative modeling, starting with density estimation, and then concentrating on various transport-based methods. An important shift of perspective, leading to alternative methodologies, follows from understanding the role of the Langevin equation in generating samples from a measure with probability density $\Upsilon$ :

$$\frac{du}{dt} = D\log\Upsilon + \sqrt{2}\frac{dW}{dt} \tag{13.17}$$

with $u(0) \sim \zeta$. Under mild tail and smoothness assumptions on $\Upsilon$ this equation is ergodic: for suitable classes of test function $\varphi : \mathbb{R}^d \to \mathbb{R}$, we have almost surely that

$$\lim_{T\to\infty}\frac{1}{T}\int_0^T \varphi\big(u(t)\big)\,dt = \mathbb{E}^{u\sim\Upsilon}\big[\varphi(u)\big].$$

This suggests sampling via use of the gradient of its log-density $D\log\Upsilon$, which is known as the *score function*. Learning the score function and solving the Langevin equation approximately provides one such method, but other variants on this are available and we now present some of them.

We would like to find a function $s_\theta\colon \mathbb{R}^d \to \mathbb{R}^d$ so that $s_\theta \approx D\log\Upsilon$. We are given only samples from $\Upsilon$: we do not assume access to the target density $\Upsilon$ itself. We assume that $\theta$ is chosen from set $\Theta \subseteq \mathbb{R}^p$. A natural objective results in the following regression problem:

$$\mathsf{J}(\theta;\Upsilon) = \int |s_\theta(u) - D\log\Upsilon(u)|^2\Upsilon(u)\,du, \tag{13.18a}$$

$$\theta^\star \in \arg\min_{\theta\in\Theta}\mathsf{J}(\theta). \tag{13.18b}$$

Then $s^\star = s_{\theta^\star}$ is the approximate score at the optimal $\theta$.

Remark 13.10. Let $\Upsilon_\theta$ be a distribution whose score is given by $s_\theta = D\log\Upsilon_\theta$. Then, the objective $\mathsf{J}$ corresponds to the squared *Fisher divergence*

$$d_F(\Upsilon\|\Upsilon_\theta)^2 := \int |D\log\Upsilon_\theta(u) - D\log\Upsilon(u)|^2\Upsilon(u)\,du.$$

♢

This approach looks attractive, but since the true score is not available, approximate evaluation of the objective $\mathsf{J}$, directly employing (13.18), is not feasible in practice. However, the following two propositions show that $\mathsf{J}$ can be minimized using a score-matching procedure that uses integration by parts to redefine the objective into one that only depends on $\Upsilon$ in an outer expectation that can be readily approximated using samples from $\Upsilon$.

**Proposition 13.11.** *Assume the score of* $\Upsilon$ *is square-integrable, i.e.,* $\int |D\log\Upsilon(u)|^2\Upsilon(u)\,du < \infty$*. Then, the optimal score that minimizes* $\mathsf{J}(\theta;\Upsilon)$ *is given by* $s^\star = s_{\theta^\star}$ *where*

$$\theta^\star \in \arg\min_{\theta\in\Theta}\int\Big(|s_\theta(u)|^2 + 2\mathrm{Tr}\big(Ds_\theta(u)\big)\Big)\Upsilon(u)\,du. \tag{13.19}$$

*Proof.* Expanding the Euclidean norm, the least-squares objective (13.18) yields

$$\mathsf{J}(\theta;\Upsilon) = \int \Big(|s_\theta(u)|^2 - 2\langle s_\theta(u), D\log\Upsilon(u)\rangle + |D\log\Upsilon(u)|^2\Big)\Upsilon(u)\,du.$$

Using integration by parts and that $\Upsilon$ is a density which decays to zero as $|u| \to \infty$, we can write the second term as

$$\int \langle s_\theta(u), D\log\Upsilon(u)\rangle \Upsilon(u)\,du = \int \langle s_\theta(u), D\Upsilon(u)\rangle\,du = -\int \mathrm{Tr}\big(Ds_\theta(u)\big)\Upsilon(u)\,du.$$

We can thus rewrite the objective $\mathsf{J}$ as

$$\begin{aligned}\mathsf{J}(\theta;\Upsilon) &= \int \Big(|s_\theta(u)|^2 + 2\mathrm{Tr}\big(Ds_\theta(u)\big)\Big)\Upsilon(u)\,du + K,\\ K &= \int |D\log\Upsilon(u)|^2\Upsilon(u)\,du.\end{aligned}$$

Since $K$ is independent of $\theta$ and finite by assumption, we deduce that the minimizers of $\mathsf{J}(\theta;\Upsilon)$ coincide with those defined in (13.19). □

Remark 13.12. We note that, in contrast to (13.18), the objective (13.19) requires evaluation of the gradient of the approximate score function $Ds_\theta$, which is a Hessian matrix of size $d \times d$. This matrix can be challenging to compute and store in high-dimensions. However, one may leverage techniques from randomized linear algebra to estimate the trace of this matrix. ♢

The need to evaluate the Hessian can also be avoided by use of *denoising score-matching*. This approach estimates the score function for an approximate distribution defined by the convolution of $\Upsilon$ with a kernel $p_\sigma\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ that has bandwidth $\sigma > 0$. That is,

$$\Upsilon_\sigma(u) \coloneqq \int p_\sigma(u,w)\Upsilon(w)\,dw. \tag{13.21}$$

For example, $p_\sigma$ can be a Gaussian kernel

$$p_\sigma(w,u) \propto \exp\Big(-\frac{1}{2\sigma^2}|w-u|^2\Big).$$

This choice is motivated by the facts that: (i) for bandwidth $\sigma \to 0$ we obtain that $p_\sigma$ converges to a Dirac measure and so $\Upsilon_\sigma(u)$ converges to $\Upsilon(u)$, the original distribution; (ii) on the other hand, larger bandwidths smooth larger-scale features of $\Upsilon$. The following proposition shows that score matching for $\Upsilon_\sigma$ can be performed without knowing the score of $\Upsilon$ directly.

**Proposition 13.13.** *Assume that*

$$\int |D\log\Upsilon_\sigma(u)|^2\Upsilon_\sigma(u)\,du + \int\int |D_u \log p_\sigma(u,w)|^2\Upsilon_\sigma(w,u)\Upsilon(u)\,dwdu < \infty.$$

*Then, the optimal score that minimizes* $\mathsf{J}(\theta;\Upsilon_\sigma)$ *is given by* $s^\star = s_{\theta^\star}$ *where*

$$\theta^\star \in \arg\min_{\theta\in\Theta} \int\int |s_\theta(u) - D_u\log p_\sigma(u,w)|^2 p_\sigma(u,w)\Upsilon(w)\,dudw. \tag{13.22}$$

*Proof.* The score of the smoothed distribution $p_\sigma$ is given by

$$\begin{aligned} D\log\Upsilon_\sigma(u) = \frac{D\Upsilon_\sigma(u)}{\Upsilon_\sigma(u)} &= \frac{1}{\Upsilon_\sigma(u)}\int D_u p_\sigma(u,w)\Upsilon(w)\,dw \\ &= \frac{1}{\Upsilon_\sigma(u)}\int D_u(\log p_\sigma(u,w))p_\sigma(u,w)\Upsilon(w)\,dw. \end{aligned}$$

Thus, the inner product of $s_\theta$ and the score of the smoothed distribution is given by

$$\int \langle s_\theta(u), D\log\Upsilon_\sigma(u)\rangle\Upsilon_\sigma(u)\,du = \int\int \langle s_\theta(u), D_u\log p_\sigma(u,w)\rangle p_\sigma(u,w)\Upsilon(w)\,dwdu.$$

Furthermore

$$\int |s_\theta(u)|^2\Upsilon_\sigma(u)\,du = \int\int |s_\theta(u)|^2 p_\sigma(u,w)\Upsilon(w)\,dwdu$$

by definition of $\Upsilon_\sigma$. Substituting the two preceding expressions into the objective $\mathsf{J}(\theta;\Upsilon_\sigma)$ gives

$$\begin{aligned} \mathsf{J}(\theta;\Upsilon_\sigma) &= \int\int |s_\theta(u) - D_u\log p_\sigma(u,w)|^2 p_\sigma(u,w)\Upsilon(w)\,dudw + K, \\ K &= \int |D\log\Upsilon_\sigma(u)|^2\Upsilon_\sigma(u)\,du - \int\int |D_u\log p_\sigma(u,w)|^2 p_\sigma(u,w)\Upsilon(w)\,dudw. \end{aligned}$$

Noting that $K$ is independent of $\theta$ and finite by assumption, we deduce that the minimizers of $\mathsf{J}(\theta;\Upsilon_\sigma)$ coincide with those defined in (13.22). □

The score of smoothed distribution $\Upsilon_\sigma$ that is computed by Proposition 13.13 can also be expressed in terms of a conditional expectation of the underlying random variables. The next lemma presents this result, which is known as *Tweedie's formula*.

**Lemma 13.14.** *Let $u = w + \sigma\eta$ where $w \sim \Upsilon$ and $\eta \sim \mathcal{N}(0, I_d)$. Define $\pi^w$ to be the posterior density*[2]

$$\pi^w(w|u) = \frac{1}{\Upsilon_\sigma(u)}p_\sigma(u,w)\Upsilon(w),$$

*where $p_\sigma$ is the Gaussian kernel. Then the score $D\log\Upsilon_\sigma(u)$ satisfies the identity*

$$\mathbb{E}^{w\sim\pi^w}[w] = u + \sigma^2 D\log\Upsilon_\sigma(u). \tag{13.23}$$

*Proof.* First note that the score of the Gaussian kernel is

$$D_u\log p_\sigma(u,w) = -\frac{(u-w)}{\sigma^2}.$$

Using this we can write the score of the smoothed distribution as

$$\begin{aligned} D\log\Upsilon_\sigma(u) &= \frac{1}{\Upsilon_\sigma(u)}\int D_u(\log p_\sigma(u,w))p_\sigma(u,w)\Upsilon(w)\,dw \\ &= \int \frac{-(u-w)}{\sigma^2}\frac{p_\sigma(u,w)\Upsilon(w)}{\Upsilon_\sigma(u)}\,dw \\ &= \frac{1}{\sigma^2}\mathbb{E}^{w\sim\pi^w}[w-u], \end{aligned}$$

and the desired result follows. □

[2] Note that this is the correct normalization by (13.21).

Tweedie's formula shows that computing a score function is related to an inverse problem. It may be rearranged to give:

$$D\log\Upsilon_\sigma(u)=\frac{1}{\sigma^2}\Big(\mathbb{E}^{w\sim\pi^w}[w]-u\Big).$$

Given a noisy realization $u$ of the random variable $w$, the score is given in terms of the conditional expectation $\mathbb{E}[w|u]$, which is the posterior mean for the unknown $w$. Notice that $\mathbb{E}[w|u]$ is the function of $u$ that minimizes $\mathbb{E}|w-\varphi(u)|^2$ over measurable maps $\varphi:\mathbb{R}^d\to\mathbb{R}^d$. Thus, to learn $D\log\Upsilon_\sigma(u)$ we can seek parameterized map $\varphi_\theta:\mathbb{R}^d\times\Theta\to\mathbb{R}^d$ that minimizes $\mathbb{E}|w-\varphi_\theta(u)|^2$. This problem can be empirically approximated without having access to the density $\Upsilon$. Specifically, given only data $\{w^{(n)}\}_{n=1}^N$ independently sampled from $\Upsilon$, we set $u^{(n)}:=w^{(n)}+\sigma\eta^{(n)}$, where $\eta^{(n)}\sim\mathcal{N}(0,I_d)$ i.i.d. We then set $\theta^\star\in\arg\min_\theta\sum_{n=1}^N|w^{(n)}-\varphi_\theta(u^{(n)})|^2$, define $\varphi^\star=\varphi_{\theta^\star}$ and approximate $D\log\Upsilon_\sigma(u)\approx\frac{1}{\sigma^2}\big(\varphi^\star(u)-u\big)$.

## 13.5 Autoencoders

Autoencoders are primarily used as a technique for uncovering latent low-dimensional structure in high dimensional data. We introduce them here and then demonstrate in the next section that a natural probabilistic relaxation of the idea leads to generative models termed variational autoencoders.

Transport ideas may be used to (approximately) represent a complicated probability distribution, perhaps only known through samples, as the pushforward of a simpler measure. In this section we go a step further, by seeking a simpler measure which lives in a lower dimensional latent space. Autoencoders are one natural approach to such problems.

The basic idea of autoencoders is to find an approximate factorization of the identity which is accurate in the support of density $\Upsilon$, but using only the empirical approximation of $\Upsilon$ by $\Upsilon^{\mathrm{N}}$. Let $f:\mathbb{R}^d\times\Theta_f\to\mathbb{R}^{d_z}$ and $g:\mathbb{R}^{d_z}\times\Theta_g\to\mathbb{R}^d$ where $\Theta_f\subseteq\mathbb{R}^{p_f}$ and $\Theta_g\subseteq\mathbb{R}^{p_g}$; in particular each of $f$ and $g$ can be a neural network, as defined in Section 14.1, generalized to vector-valued output. Recalling that Id denotes the identity mapping on $\mathbb{R}^d$, define[3]

$$\mathsf{J}(\theta_f,\theta_g)=\Big\|\mathrm{Id}-g(f(\cdot;\theta_f);\theta_g)\Big\|_{H_\Upsilon}^2:=\mathbb{E}^{u\sim\Upsilon}\big|u-g\big(f(u;\theta_f);\theta_g\big)\big|^2.$$

Now consider the following optimization problem:

$$(\theta_f^\star,\theta_g^\star)\in\arg\min_{(\theta_f,\theta_g)\in\Theta_f\times\Theta_g}\mathsf{J}(\theta_f,\theta_g).\tag{13.24}$$

We then define $f^\star(u)=f(u;\theta_f^\star)$ and $g^\star(z)=g(z;\theta_g^\star)$. Roughly speaking, and dropping the dependence on parameters for expository purposes, we are seeking functions $f$ and

[3] The notation $H_\Upsilon$ is used in the next chapter, equation (14.2), in the context of real-valued functions and is readily generalized to $\mathbb{R}^d-$valued functions.

$g$ such that $g\big(f(u)\big) \approx u$. In practice the optimization is implemented empirically, and we minimize

$$\mathsf{J}^N(\theta_f,\theta_g) = \Big\|\mathrm{Id} - g\big(f(\cdot;\theta_f);\theta_g\big)\Big\|^2_{H^N_\Upsilon} = \frac{1}{N}\sum_{n=1}^N \big|u^{(n)} - g\big(f(u^{(n)};\theta_f);\theta_g\big)\big|^2.$$

Remark 13.15. This approach reduces the autoencoder to a particular form of supervised learning in which the inputs and outputs are equal, so we seek to approximate the identity, and in which we impose a specific structure on the class of approximating function – as composition of $g$ with $f$. We refer to $\mathbb{R}^{d_z}$ as the *latent space* and note that a primary application of the methodology is to identify latent spaces of dimension $d_z$ which is much less than the dimension $d$ of the data space. The approximate factorization of the identity found by composing $g$ with $f$ provides a way of moving between the data space and the latent space. ♢

**Example 13.16.** We demonstrate that *principal component analysis*, often referred to simply as PCA, may be viewed as a form of autoencoder. Define

$$\begin{aligned} m &= \mathbb{E}^{u\sim\Upsilon}u,\\ C &= \mathbb{E}^{u\sim\Upsilon}\Big[(u-m)\otimes(u-m)\Big]. \end{aligned}$$

Then consider the eigenpairs $(\varphi^{(i)},\lambda^{(i)}) \in \mathbb{R}^d \times \mathbb{R}^+$ solving

$$\begin{aligned} C\varphi^{(i)} &= \lambda^{(i)}\varphi^{(i)},\\ |\varphi^{(i)}| &= 1. \end{aligned}$$

The eigenvalues are positive and we assume them to be decreasingly ordered with respect to $i$. Moreover, the eigenvectors are orthonormal.

For given $d_z \le d$, we now define the maps

$$\begin{aligned} f(u) &= \Big(\langle u,\varphi^{(1)}\rangle,\cdots,\langle u,\varphi^{(d_z)}\rangle\Big),\\ g(z) &= \sum_{j=1}^{d_z} z_j\varphi^{(j)}. \end{aligned}$$

Then, their composition gives us

$$\begin{aligned} g\big(f(u)\big) &= \sum_{j=1}^{d_z}\langle u,\varphi^{(j)}\rangle\varphi^{(j)}\\ &= \Big(\sum_{j=1}^{d_z}\varphi^{(j)}\otimes\varphi^{(j)}\Big)u. \end{aligned}$$

Thus, the PCA method may be viewed as truncating the representation of the identity defined by the spectral theorem for positive, symmetric matrices. The specific basis used is defined by $\Upsilon$ but it is approximated in practice knowing only $\Upsilon^{\mathrm{N}}$ by computing the empirical mean and covariance. ♢

## 13.6 Variational Autoencoders

In the preceding section, note that $\zeta = (f^\star)_\sharp \Upsilon$ gives the (approximate) distribution in the latent space, but that we cannot specify $\zeta$; mapping $g$ (approximately) solves a measure transport problem from $\zeta$ to $\Upsilon$, but choice of measure $\zeta$ is not within our control. It is natural to ask whether the methodology can be generalized to settings in which it is desirable to impose a specified distribution $\zeta$. This leads to the topic of *variational autoencoders.*

Let $\pi$ be a coupling of $\Upsilon$ and $\zeta$ and note that the following two identities hold:

$$\pi(u,z) = \mathbb{P}(u|z)\zeta(z),$$
$$\pi(u,z) = \mathbb{P}(z|u)\Upsilon(u).$$

The idea of the variational autoencoder is to approximate the two conditional densities $\mathbb{P}(u|z), \mathbb{P}(z|u)$ appearing in these identities by parameterized families. A parameter choice is made to ensure that the two resulting approximate expressions for $\pi(u,z)$ are close. To be explicit we assume that

$$u|z \sim \mathcal{N}(g(z;\theta_g), \sigma_g^2 I),$$
$$z|u \sim \mathcal{N}(f(u;\theta_f), \sigma_f^2 I),$$

noting that this is a relaxation of the setting for autoencoders. Invoking these Gaussian approximations for the conditionals we obtain the following two approximations for the coupling density:

$$\pi_g(u,z) = \frac{1}{Z_g}\exp\Bigl(-\frac{1}{2\sigma_g^2}|u-g(z;\theta_g)|^2\Bigr)\zeta(z),$$
$$\pi_f(u,z) = \frac{1}{Z_f}\exp\Bigl(-\frac{1}{2\sigma_f^2}|z-f(u;\theta_f)|^2\Bigr)\Upsilon(u).$$

A common choice is to assume that $\zeta$ is the density of a standard unit Gaussian; we make this assumption throughout what follows.

For simplicity, we consider the standard deviations $\sigma_f, \sigma_g$ to be fixed, and determine the parameters of $f$ and $g$. We do this by asking that $\pi_f$ and $\pi_g$ are close. If we measure closeness by means of KL divergence, with $\pi_f$ in the first argument, then using the Gaussian assumptions on the conditionals and on $\zeta$ leads to insightful explicit calculations. To see this, we first note that

$$\mathsf{D}_{\mathrm{KL}}(\pi_f\|\pi_g) = \mathbb{E}^{u\sim\Upsilon}\Bigl[\mathbb{E}^{z\sim\mathbb{P}(z|u)}\Bigl[\frac{1}{2\sigma_g^2}|u-g(z;\theta_g)|^2\Bigr] + \mathsf{D}_{\mathrm{KL}}\bigl(\mathbb{P}(\cdot|u)\|\zeta\bigr)\Bigr] + \text{const},$$

where the constant is independent of the parameters we wish to learn and where $\mathbb{P}(z|u)$ and $\zeta(z)$ are given by their assumed Gaussian structure. Using the expression in Example 12.30 for the KL divergence between two Gaussians we obtain

$$\mathsf{D}_{\mathrm{KL}}(\pi_f\|\pi_g) = \mathbb{E}^{u\sim\Upsilon}\Bigl[\mathbb{E}^{z\sim\mathcal{N}\left(f(u;\theta_f),\sigma_f^2 I\right)}\Bigl[\frac{1}{2\sigma_g^2}|u-g(z;\theta_g)|^2\Bigr] + \frac{1}{2}|f(u;\theta_f)|^2\Bigr] + \text{const}.$$

Noting that $z \sim \mathcal{N}(f(u;\theta_f), \sigma_f^2 I)$ is the same in law as $f(u;\theta_f)+\sigma_f \xi$ where $\xi \sim \mathcal{N}(0,I)$, we may write this divergence as

$$\mathsf{D}_{\mathrm{KL}}(\pi_f \| \pi_g) = \mathsf{J}(\theta_f, \theta_g) + \mathrm{const},$$

where

$$\mathsf{J}(\theta_f,\theta_g) = \mathbb{E}^{u\sim\Upsilon}\Big[\mathbb{E}^{\xi\sim\mathcal{N}(0,I)}\Big[\frac{1}{2\sigma_g^2}\Big|u - g\big(f(u;\theta_f)+\sigma_f\xi;\theta_g\big)\Big|^2\Big] + \frac{1}{2}|f(u;\theta_f)|^2\Big]$$

and the constant term is independent of $(\theta_f, \theta_g)$.

This constitutes a regularized version of the standard autoencoder; in particular the preceding expression for the divergence clearly regularizes the basic concept that $g(f(u)) \approx u$ under density $\Upsilon$.

Remark 13.17. In practice this is implemented with expectation over $\Upsilon$ approximated empirically by $\Upsilon^{\mathrm{N}}$. This leaves an optimization problem in which the objective is defined via expectation over $\xi$. Stochastic gradient descent from Chapter 16 may be used to tackle this problem. After training, approximate samples from $\Upsilon$ may be obtained by sampling $z \sim \zeta$ and then sampling $u|z \sim \mathcal{N}(g(z;\theta_g), \sigma_g^2)$. ♢

## 13.7 Generative Adversarial Networks

The variational autoencoder trains a probabilistic model for $u$ by learning $\mathbb{P}(u|z)$ and specifying $\mathbb{P}(z)$. The generative adversarial network is another method for training such a probabilistic model, which also allows for low dimensional latent space. To be concrete we assume that it has the same structural form, namely

$$\begin{aligned} u|z &\sim \mathcal{N}(g(z;\theta_g), \sigma_g^2 I),\\ z &\sim \mathcal{N}(0,I). \end{aligned}$$

Thus samples from $u$ are created from samples from $z$. We write the resulting density of $u$ as $\Upsilon_g$, with $g$ for generative. It is desired that samples from $\Upsilon_g$ are similar to those specified by the data, which is drawn from density $\Upsilon$.

The generative adversarial network starts by defining a *discriminator* $\mathsf{d} : \mathbb{R}^d \times \Theta_{\mathsf{d}} \to [0,1]$, to be thought of as taking values which are probabilities. Here $\Theta_{\mathsf{d}} \subset \mathbb{R}^{p_{\mathsf{d}}}$. It is instructive to think of $\mathsf{d}(u;\theta_{\mathsf{d}})$ as being the probability that $u$ is drawn from the density $\Upsilon$, and $1-\mathsf{d}(u;\theta_{\mathsf{d}})$ as the probability of $u$ being drawn from the generative model $\Upsilon_g$. We then define $v : \Theta_g \times \Theta_{\mathsf{d}} \to \mathbb{R}$ by

$$v(\theta_g, \theta_{\mathsf{d}}) = \mathbb{E}^{u\sim\Upsilon}\Big[\log \mathsf{d}(u;\theta_{\mathsf{d}})\Big] + \mathbb{E}^{u\sim\Upsilon_g}\Big[\log(1-\mathsf{d}(u;\theta_{\mathsf{d}}))\Big].$$

The optimal parameter values are defined by

$$\begin{aligned} \widetilde{\theta}_{\mathsf{d}}(\theta_g) &\in \arg\max_{\theta_{\mathsf{d}}} v(\theta_g, \theta_{\mathsf{d}}),\\ \theta_g^\star &\in \arg\min_{\theta_g} v(\theta_g, \widetilde{\theta}_{\mathsf{d}}(\theta_g)),\\ \theta_{\mathsf{d}}^\star &= \widetilde{\theta}_{\mathsf{d}}(\theta_g^\star). \end{aligned}$$

In the maximization step, the parameters of the discriminator are chosen to maximize $v$ – the discriminator acts adversarially to try and find data under $\Upsilon_g$ which does not look like data under $\Upsilon$. In the minimization step the generator acts to reduce $v$ to try and make the two data sources look similar. Ideally, through an iterative process, a solution is found in which $\mathsf{d}(\cdot;\theta_{\mathsf{d}}^{\star}) \equiv \frac{1}{2}$, so that the data and generated data are indistinguishable. The value of $\theta_g^{\star}$ defines the generative model once this indistinguishable state has been reached.

## 13.8 Bibliography

Density estimation is covered in numerous texts; we refer the reader to the book [288] and the literature cited there. For a foundational reference concerning the use of normalizing flows in machine learning see [270]; more recent applications may be found in [335], [101]. The neural ODE perspective was popularized in the paper [55]; earlier papers [125, 87] anticipated the idea but did not make a practical tool in the way that the neural ODE paper [55] did. The subject of optimal transport is overviewed and systematically developed in [320]; computational methodology is overviewed and developed in [249]. Triangular transport maps are overviewed as a method for sampling in [212].

Score-based methods have become popular in the context of generative diffusion models [293]. The integration by parts method and denoising score matching approaches were first proposed in [156] and [321], respectively. These approximate scores have been used for sampling using Langevin dynamics; see [191, 31] for sampling guarantees. They have also been used in other unsupervised learning tasks, like learning probability graphical models. Autoencoders have a long history; see [118, 141, 286] and the citations therein. Variational autoencoders were introduced concurrently in [173] and [271]. For overviews of variational autoencoders see [339, 174]. The idea of conducting unsupervised learning with GANs was introduced in [117].

Although not covered here, because our focus is on generative modeling, clustering is a widely used methodology in unsupervised learning; it is overviewed in [324]. The paper [232] describes an underlying mathematical framework, based on eigenvalue perturbation theory. Understanding large data limits of spectral clustering methods is a subject developed in [325, 147]. The use of graph Laplacians, which underpin spectral clustering, in the solution of inverse problems, is undertaken in [105, 86, 107, 131, 277, 133].

# Chapter 14

## Supervised Learning

This chapter is concerned with the *supervised learning* task of determining a function $\psi^\dagger : D \subseteq \mathbb{R}^d \to \mathbb{R}$ from data in the form of input-output pairs from the map. The codomain of $\psi^\dagger$ may have finite cardinality (classification) or infinite cardinality (regression). We focus on regression since this task arises naturally in the context of learning forward maps for inverse problems and dynamical systems for data assimilation. Our data assumption is as follows:

**Data Assumption 14.1.** *Data is available in the form*

$$\left\{u^{(n)}, y^{(n)}\right\}_{n=1}^{N}, \tag{14.1}$$

*where the $\{u^{(n)}\}_{n=1}^N$ are generated i.i.d. from probability density function $\Upsilon(u)$, supported on $D \subseteq \mathbb{R}^d$, and where $y^{(n)} = \psi^\dagger(u^{(n)}) + \sqrt{\lambda}\xi^{(n)}$, where $\{\xi^{(n)}\}_{n=1}^N$ is an i.i.d. mean zero noise process.*

The parameter $\lambda \geq 0$ allows us to consider noiseless data ($\lambda = 0$) and noisy data ($\lambda > 0$). Since the goal is to determine $\psi^\dagger$ we are concerned with function approximation, and to that end we will study three approaches to parameterize and learn functions: neural networks (Section 14.1), random features (Section 14.2), and Gaussian processes (Section 14.3). We will concentrate on the noiseless case $\lambda = 0$, but will briefly mention the noisy case $\lambda > 0$ in the context of random features and Gaussian processes.

We let $H_\Upsilon$ denote the Hilbert space of real-valued functions on $D$ with inner-product and induced norm

$$\langle \psi, \varphi \rangle_{H_\Upsilon} = \int_D \psi(u)\varphi(u)\Upsilon(u)\, du, \quad |\psi|_{H_\Upsilon}^2 = \langle \psi, \psi \rangle_{H_\Upsilon}.$$

Thus

$$|\psi|_{H_\Upsilon} = \left(\int_D \psi(u)^2 \Upsilon(u)\, du\right)^{1/2}. \tag{14.2}$$

In practice, the probability density function $\Upsilon$ is typically unknown but may be approximated by the empirical density $\Upsilon^{\mathrm{N}}$ of the data $\{u^{(n)}\}_{n=1}^N$, given by

$$\Upsilon^{\mathrm{N}}(u) = \frac{1}{N}\sum_{n=1}^{N} \delta(u - u^{(n)}).$$

We may then define an empirical approximation of $|\psi|_{H_\Upsilon}$ by replacing $\Upsilon$ by $\Upsilon^{\mathrm{N}}$ in (14.2) to obtain

$$|\psi|_{H_{\Upsilon^{\mathrm{N}}}} = \left(\frac{1}{N}\sum_{n=1}^{N} \psi(u^{(n)})^2\right)^{1/2}.$$

## 14.1 Neural Networks

A neural network is a parametric family of functions. By use of the data a value of the parameter may be chosen so as to determine an approximation of $\psi^\dagger$. For simplicity we work under Data Assumption 14.1 in the setting $\lambda = 0$. To define a neural network we first define:

**Definition 14.2.** An *activation function* $\sigma : \mathbb{R} \to \mathbb{R}$ is a monotonic non-decreasing function. It is extended to $\sigma : \mathbb{R}^s \to \mathbb{R}^s$ pointwise: $\sigma(u)_m = \sigma(u_m)$ for $u = (u_1, \cdots, u_m, \cdots, u_s)$ and $\sigma(u) = (\sigma(u)_1, \cdots, \sigma(u)_m, \cdots, \sigma(u)_s)$. ♢

**Example 14.3.** The *Rectified Linear Unit (ReLU)* activation function is $\sigma(u) = \max(u, 0)$. The *Gaussian Error Linear Unit (GELU)* activation function is $\sigma(u) = uF(u)$, where $F$ is the cumulative distribution function of the scalar unit centred Gaussian. The *Scaled Exponential Linear Unit (SELU)* activation function is

$$\sigma(u) = \begin{cases} \kappa u, & u > 0, \\ \kappa\alpha(\exp(u) - 1), & u \le 0. \end{cases} \tag{14.3}$$

The *Exponential Linear Unit (ELU)* activation function is obtained from the SELU by setting $\kappa = 1$. ♢

Now introduce $\Theta \subseteq \mathbb{R}^{d_\theta}$ and then $\psi : \mathbb{R}^d \times \Theta \to \mathbb{R}$ via the iteration

$$\psi_0(u;\theta) = u, \tag{14.4}$$

$$\psi_{\ell+1}(u;\theta) = \sigma(W_\ell \psi_\ell(u;\theta) + b_\ell), \quad \ell = 0, \dots, L-1, \tag{14.5}$$

$$\psi(u;\theta) = \beta^\top \psi_L(u;\theta), \tag{14.6}$$

where, for $\ell \in \{0, \dots, L-1\}$, $W_\ell \in \mathbb{R}^{d_{\ell+1}\times d_\ell}$ and $b_\ell \in \mathbb{R}^{d_{\ell+1}}$, with $d_0 = d$ and $\beta \in \mathbb{R}^{d_L}$; together the matrices $\{W_\ell\}_{\ell=0}^{L-1}$ and vectors $\{b_\ell\}_{\ell=0}^{L-1}$ define $\theta$ (and $d_\theta$). Indeed here we define $\vartheta = \{W_\ell, b_\ell\}_{\ell=0}^{L-1}$ and $\theta = (\vartheta, \beta)$.

Resulting function $\psi : \mathbb{R}^d \times \Theta \to \mathbb{R}$ is known as a *(deep*[1]*) neural network*. The reader will note that the concept is readily generalized to $\psi : \mathbb{R}^d \times \Theta \to \mathbb{R}^q$ for any integer $q$; we concentrate on $q = 1$ for simplicity of exposition only.

An idealized approach to determining $\theta$ is to minimize the *risk*

$$\mathsf{J}(\theta) := |\psi^\dagger - \psi(\cdot;\theta)|^2_{H_\Upsilon}$$

[1] Deep if $L$ is large enough; often 3 or more is considered large in this context. Lower values are sometimes referred to as *shallow neural networks*.

by setting

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathsf{J}(\theta). \tag{14.7}$$

However, this requires knowing $\psi^\dagger$, which is of course not given, as well as the probability density function $\Upsilon$. Instead the optimal parameter $\theta^\star$ is chosen to minimize the *empirical risk*

$$\mathsf{J}^N(\theta) := |\psi^\dagger - \psi(\cdot;\theta)|^2_{H_{\Upsilon^{\mathrm{N}}}}.$$

Notice that

$$\mathsf{J}^N(\theta) = \frac{1}{N}\sum_{n=1}^{N} |y^{(n)} - \psi(u^{(n)};\theta)|^2. \tag{14.8}$$

Remark 14.4. We have derived this loss function $\mathsf{J}^N(\cdot)$ under Data Assumption 14.1 with $\lambda = 0$. However it may also be derived under the more general assumption that data set $(u^{(n)}, y^{(n)})$ is drawn i.i.d from a probability measure $\pi$ on $\mathcal{P}(\mathbb{R}^d \times \mathbb{R}^k)$. Examples of such measures may be found from Data Assumption 14.1 for any $\lambda \geq 0$, but the setting is much more general. The risk may then be written as

$$\mathsf{J}(\theta) := \mathbb{E}^{(u,y)\sim\pi}\, |y - \psi(u;\theta)|^2$$

and approximated empirically to again obtain (14.8). ♢

Minimizing (14.8) leads to an implementable strategy for function approximation. The empirical risk is typically non-convex as a function of $\theta$, with multiple saddle points and local minima; it is observed that minimization via the use of stochastic gradient descent (Chapter 16) works well for many problems. Once $\theta^\star$ is learned, we write $\psi^\star(u) := \psi(u;\theta^\star)$.

## 14.2 Random Features

We continue to work under Data Assumption 14.1 in the setting $\lambda = 0$. Now consider a thought experiment in which, rather than optimizing over all of $\Theta$, the parameters $\vartheta = \{W_\ell, b_\ell\}_{\ell=0}^{L-1}$ are fixed and optimization is performed only over $\beta$. Note that $\psi_L$ depends only on $\vartheta$ and not $\beta$; indeed $\psi_L(\cdot;\vartheta) : \mathbb{R}^d \to \mathbb{R}^{d_L}$ and we may define $\varphi_i(\cdot;\vartheta) : \mathbb{R}^d \to \mathbb{R}$ to be the $i^{th}$ component, for $i = 1, \dots, d_L$, of the vector-valued output function. We can then write

$$\psi(u;\vartheta,\beta) := \sum_{i=1}^{d_L} \beta_i \varphi_i(u;\vartheta), \tag{14.9}$$

viewing $\psi(\cdot;\vartheta,\beta)$ as parameterized by $\beta$, since the remaining elements $\vartheta$ of $\theta$ have been fixed. From $\psi^\dagger$ we may define

$$\begin{aligned}\beta^\star &\in \arg\min_{\beta\in\mathbb{R}^{d_L}} |\psi^\dagger - \psi(\cdot;\vartheta,\beta)|^2_{H_{\Upsilon^{\mathrm{N}}}}\\ &= \arg\min_{\beta\in\mathbb{R}^{d_L}} \sum_{n=1}^{N} |y^{(n)} - \psi(u^{(n)};\vartheta,\beta)|^2\end{aligned}$$

and $\psi^\star(u) := \psi(u;\vartheta,\beta^\star)$. In contrast to full optimization over $\theta$, this leads to a convex, indeed quadratic, optimization problem that can be readily solved.

However, we have simply fixed $\vartheta$; we have not discussed how to choose it. One idea is to simply pick $\vartheta$ at random from some probability distribution. With our current construction, the resulting collection of functions $\{\varphi_i\}$ are then random but not, in general, i.i.d. A variant on this idea is to choose $q \in \mathcal{P}(\mathbb{R}^{d_\vartheta})$, define $\varphi : \mathbb{R}^d \times \mathbb{R}^{d_\vartheta} \to \mathbb{R}$ and set

$$\psi(u;\beta) := \sum_{i=1}^{d_L} \beta_i \varphi(u;\vartheta_i), \quad \vartheta_i \sim q, \text{ i.i.d.} \tag{14.10}$$

**Example 14.5.** Random Fourier features take the form

$$\varphi(u;\vartheta) = \cos\big(\langle \omega, u\rangle + b\big)$$

where $\vartheta = (\omega, b)$ is chosen at random. It is natural to take $b \sim U[0,2\pi]$, and we assume $\omega \sim W$, for some $W \in \mathcal{P}(\mathbb{R}^d)$. ♢

**Example 14.6.** Each $\varphi(\cdot;\vartheta_i)$ may be chosen as a neural network, for example, with the parameters of that neural network chosen i.i.d. at random from given distribution $q$. ♢

Empirical risk minimization to determine vector $\beta$ in (14.10) again leads to a linear system for $\beta$ found by minimizing a quadratic loss function. This loss function may be regularized and then $\beta^\star$ is defined via

$$\beta^\star \in \arg\min_{\beta\in\mathbb{R}^{d_L}} \mathsf{J}^N(\beta), \tag{14.11}$$

$$\mathsf{J}^N(\beta) := \frac{1}{2}\sum_{n=1}^{N} |y^{(n)} - \psi(u^{(n)};\beta)|^2 + \frac{\lambda}{2}|\beta|^2. \tag{14.12}$$

Similarly to before we write $\psi^\star(u) := \psi(u;\beta^\star)$.

Remark 14.7. The regularization is quadratic and hence does not change the simplicity of the optimization problem for $\beta$ : it still results in solution of a linear system. Furthermore the specific regularization arises naturally in the context of noisy data $\xi^{(n)} \sim \mathcal{N}(0,I)$ in (14.1) – see Remark 14.10. ♢

We refer to the functions $\varphi(\cdot;\vartheta)$, with $\vartheta \sim q$, as *random features*. It is natural to ask how correlated these functions are at different points in $D$. To this end, we introduce a *kernel* $c : D \times D \to \mathbb{R}$ defined by

$$c(u,u') := \mathbb{E}^q\big[\varphi(u;\vartheta)\varphi(u';\vartheta)\big], \tag{14.13}$$

where expectation is over $\vartheta \sim q$.

**Example 14.8.** Consider Example 14.5. We note that

$$\begin{aligned} c(u,u') &= \int I(\omega)W(\omega)\,d\omega,\\ I(\omega) &= \frac{1}{2\pi}\int_0^{2\pi} \cos\big(\langle\omega,u\rangle + b\big)\cos\big(\langle\omega,u'\rangle + b\big)\,db\\ &= \frac{1}{2}\cos\big(\langle\omega,u-u'\rangle\big). \end{aligned}$$

♢

## 14.3 Gaussian Processes

### 14.3.1 Kernels

The theory of Gaussian process regression starts from a kernel satisfying two key properties:

**Definition 14.9.** Let $D \subseteq \mathbb{R}^d$. A kernel $c : D \times D \to \mathbb{R}$ is *symmetric* if $c(u,u') = c(u',u)$ for all $(u,u') \in D \times D$. It is *non-negative definite* if, for all $N \in \mathbb{N}$, all $\{u^{(i)}\}_{i=1}^N \subset D$ and all $e \in \mathbb{R}^N$,

$$\sum_{i=1}^N \sum_{j=1}^N c(u^{(i)}, u^{(j)}) e_i e_j \geq 0.$$

If equality implies that $e = 0$, the kernel is called *positive definite.* ♢

A symmetric non-negative definite kernel may be viewed as a *covariance function* of a Gaussian random field. From kernel $c$, we define the integral operator

$$(\mathcal{C}\varphi)(u) = \int_D c(u,u')\varphi(u')\, du'. \tag{14.14}$$

Provided that $D$ is compact and $c$ is continuous and positive definite, $\mathcal{C}$ is a positive definite trace-class operator on $L^2(D)$ and we may define the domain of the inverse of its square root as $\mathcal{K} = \mathrm{Dom}(\mathcal{C}^{-\frac{1}{2}})$. Furthermore we define $\mathcal{K}^*$ the dual space of linear functionals on $\mathcal{K}$; thus $\mathcal{K} \subset L^2(D) \subset \mathcal{K}^*$. In fact $\mathcal{K}$ is compactly embedded into $L^2(D)$ and $\mathcal{C}^{-\frac{1}{2}}$ is densely defined on $L^2(D)$. We define

$$\langle \varphi, \psi \rangle_{\mathcal{K}} = \langle \mathcal{C}^{-\frac{1}{2}}\varphi, \mathcal{C}^{-\frac{1}{2}}\psi \rangle_{L^2(D)}$$

endowing $\mathcal{K}$ with a Hilbert space structure. Then

$$\|\varphi\|_{\mathcal{K}}^2 = |\mathcal{C}^{-\frac{1}{2}}\varphi|^2.$$

We may also define $\mathcal{C}^{-1} : \mathcal{K} \to \mathcal{K}^*$. The operator $\mathcal{C}$ defines a Gaussian measure $\mathcal{N}(0, \mathcal{C})$; the measure is supported on $L^2(D)$ since $\mathrm{Tr}(\mathcal{C}) < \infty$. The space $\mathcal{K}$ is known as the *Cameron-Martin space* associated with the Gaussian measure.

Assume now that $\mathcal{K}^*$ is rich enough that it contains Dirac measures in $D$ (and hence pointwise evaluation) and define $\mathcal{C}^{-1} : \mathcal{K} \to \mathcal{K}^*$. It follows that, for all $v \in D$,

$$\mathcal{C}^{-1} c(\cdot, v) = \delta_v(\cdot). \tag{14.15}$$

We may then note that

$$\langle \varphi, c(\cdot, v) \rangle_{\mathcal{K}} = \langle \mathcal{C}^{-\frac{1}{2}}\varphi, \mathcal{C}^{-\frac{1}{2}} c(\cdot, v) \rangle = \langle \varphi, \mathcal{C}^{-1} c(\cdot, v) \rangle = \langle \varphi, \delta_v \rangle = \varphi(v).$$

This is the reproducing property and under the conditions leading to it $\mathcal{K}$ is a *reproducing kernel Hilbert space* – RKHS for short.

### 14.3.2 Regression

We work under Data Assumption 14.1 in the setting $\lambda > 0$. We assume that $\mathcal{K}$ is an RKHS and seek to find approximation to $\psi^\dagger$, function $\psi^\star$, through the following minimization problem:

$$\psi^\star \in \arg\min_{\psi\in\mathcal{K}} \mathsf{J}^N(\psi), \tag{14.16a}$$

$$\mathsf{J}^N(\psi) := \frac{1}{2}\sum_{n=1}^{N} |y^{(n)} - \psi(u^{(n)})|^2 + \frac{\lambda}{2}\|\psi\|_{\mathcal{K}}^2. \tag{14.16b}$$

This infinite dimensional optimization problem may be viewed as a function space version of (14.8) where we minimize over functions $\psi \in \mathcal{K}$, rather than parameter $\theta \in \Theta \subset \mathbb{R}^d$. It may be derived by application of a generalization of Bayes Theorem 1.2, and then a generalization of the notion of MAP estimator from Definition 1.6, from $\mathbb{R}^d$ to $L^2(D)$.

Remark 14.10. The minimization problems (14.11) and (14.16) are connected. If the RKHS $\mathcal{K}$ in (14.16) is replaced by the RKHS

$$\mathcal{K}_{d_L} = \Big\{\psi \in \mathcal{K} : \psi(u;\beta) := \sum_{i=1}^{d_L} \beta_i \varphi(u;\vartheta_i), \quad \beta_i \in \mathbb{R}\ \forall i \in \{1,\cdots,d_L\}\Big\},$$

where $\vartheta_i \sim q$ i.i.d. then the optimization reduces to one over $\beta \in \mathbb{R}^{d_L}$ and it may be shown that this reduces to (14.11). Thus (14.11) may be viewed as approximation of (14.16) via a Monte Carlo approximation of the RKHS $\mathcal{K}$. ♢

Remark 14.11. The minimization problem (14.16) corresponds to determining the mean of a posterior distribution of an infinite dimensional Bayesian inverse problem defined as follows. We place as prior on $\psi$ a Gaussian random field with mean zero and covariance function $c(u,u')$; alternatively we may write $\mathcal{N}(0,\mathcal{C})$. The likelihood is defined by assuming noisy data in (14.1), where $\xi^{(n)} \sim \mathcal{N}(0,I)$ i.i.d. and then the negative log-likelihood is given by

$$\frac{1}{2\lambda}\sum_{n=1}^{N} |y^{(n)} - \psi(u^{(n)})|^2.$$

The posterior mean, which is also a form of MAP estimator, then satisfies (14.16). We also remark that minimization of (14.11) also has a similar interpretation as a MAP estimator, but with respect to a Gaussian process with mean zero and covariance function a random approximation of $c(u,u')$. ♢

The infinite dimensional optimization problem over $\mathcal{K}$ has the following remarkable property, which is often referred to as the *representer theorem*, and demonstrates that the optimization problem is intrinsically finite dimensional:

**Theorem 14.12.** *Function $\psi^\star$ solving (14.16) has the form*

$$\psi^\star(u) = \sum_{n=1}^{N} \alpha_n^\star c(u, u^{(n)}). \tag{14.17}$$

*Furthermore, the coefficients $\alpha^\star$ solve the following quadratic minimization problem:*

$$\alpha^\star \in \arg\min_{\alpha \in \mathbb{R}^N} \mathsf{A}^N(\alpha), \tag{14.18}$$

$$\mathsf{A}^N(\alpha) := \frac{1}{2}\sum_{r=1}^{N} \Big| y^{(r)} - \sum_{n=1}^{N} \alpha_n c(u^{(r)}, u^{(n)}) \Big|^2 + \frac{\lambda}{2} \sum_{n,r=1}^{N} \alpha_n \alpha_r c(u^{(r)}, u^{(n)}). \tag{14.19}$$

Remark 14.13. In the context of supervised learning, notice the key fact that solution to the optimization problem is defined entirely by knowledge of the kernel $c$ and by the data $\left\{u^{(n)}, y^{(n)}\right\}_{n=1}^{N}$. ♢

*Proof of Theorem 14.12.* The Euler-Lagrange equations for objective function $\mathsf{J}^N$ in (14.16) yield

$$\lambda(\mathcal{C}^{-1}\psi^\star)(u) = \sum_{n=1}^{N} (y^{(n)} - \psi^\star(u^{(n)}))\delta_{u^{(n)}}(u).$$

Now note formally from (14.15), or directly from (14.14), that $(\mathcal{C}\delta_{u'})(u) = c(u, u')$. Thus applying $\mathcal{C}$ to the Euler-Lagrange equations shows that $\psi^\star(u)$ is in the linear span of $\{c(\cdot, u^{(n)})\}_{n=1}^{N}$, establishing the first part of the result.

For the second part, let

$$\mathcal{K}^N = \Big\{\psi \in \mathcal{K} : \psi(u) = \sum_{n=1}^{N} \alpha_n c(u, u^{(n)}), \quad \alpha_n \in \mathbb{R}\ \forall n \in \{1, \cdots, N\}\Big\}$$

and define the optimization problem

$$\psi^\star \in \arg\min_{\psi \in \mathcal{K}^N} \mathsf{J}^N(\psi), \tag{14.20a}$$

$$\mathsf{J}^N(\psi) := \frac{1}{2}\sum_{n=1}^{N} |y^{(n)} - \psi(u^{(n)})|^2 + \frac{\lambda}{2}\|\psi\|_{\mathcal{K}}^2. \tag{14.20b}$$

Because of the established form (14.17) of the minimizer $\psi^\star$ of (14.16) we see that $\psi^\star$ will also be the minimizer of (14.20). Now, note that

$$\|\psi\|_{\mathcal{K}}^2 = |\mathcal{C}^{-\frac{1}{2}}\psi|_{L^2(D)}^2 = \langle \psi, \mathcal{C}^{-1}\psi\rangle_{L^2(D)}.$$

In $\mathcal{K}^N$ we have

$$\begin{aligned}\mathcal{C}^{-1}\psi(u) &= \sum_{n=1}^{N} \alpha_n \mathcal{C}^{-1} c(u, u^{(n)}) \\ &= \sum_{n=1}^{N} \alpha_n \delta_{u^{(n)}}(u).\end{aligned}$$

Thus, also in $\mathcal{K}^N$, we have, using the symmetry of $c(\cdot,\cdot)$,

$$\|\psi\|_{\mathcal{K}}^2 = \Big\langle \sum_{r=1}^N \alpha_r c(\cdot, u^{(r)}), \sum_{n=1}^N \alpha_n \delta_{u^{(n)}}(\cdot) \Big\rangle_{L^2(D)} = \sum_{n,r=1}^N \alpha_n \alpha_r c(u^{(r)}, u^{(n)})$$

as required. □

We define $C \in \mathbb{R}^{N\times N}$ to be the matrix with entries $C_{nr} = c(u^{(n)}, u^{(r)})$ and $S$ to be its inverse, with entries $S_{nr}$; notice that $C$ is positive definite (and hence invertible, so that $S$ is defined) provided that the kernel $c$ is positive definite. Then we have:

**Corollary 14.14.** *If $p^\star \in \mathbb{R}^N$ is vector with entries $p_n^\star = \psi^\star(u^{(n)})$, then $p^\star = C\alpha^\star$ and*

$$p^\star \in \arg\min_{p\in\mathbb{R}^N} \mathsf{J}^N(p), \tag{14.21}$$

$$\mathsf{J}^N(p) := \frac{1}{2}\sum_{n=1}^N |y^{(n)} - p_n|^2 + \frac{\lambda}{2}\sum_{n,r=1}^N p_n p_r S_{n,r}. \tag{14.22}$$

## 14.4 Approximation Properties

All of the preceding three methods can, in principle, approximate continuous functions to arbitrary accuracy, over a compact set. Doing so requires sufficient data. Furthermore, in the case of neural networks, this also requires sophisticated optimization techniques to obtain close to optimal solutions; in contrast, the random features and Gaussian process approaches only require solution of linear systems, resulting from a quadratic optimization problem. Typical instances of the resulting approximation theory for the three supervised learning techniques typically shows that, for given $\delta > 0$, there is a volume of data $N$ such that, with high probability with respect to the data,

$$\sup_{u\in D} |\psi^\dagger(u) - \psi^\star(u)| < \delta. \tag{14.23}$$

Theory establishing results of this form is cited in the bibliography which follows.

## 14.5 Bibliography

The subject of neural network function approximation, the core task of supervised learning, is overviewed in [80]. Universal approximation theorems may be found in [70, 250]. The use of random features was popularized as a methodology, and analyzed rigorously, in the collection of papers [258, 260, 259]. Gaussian processes are described in [334] and kernel-based methods more generally are described in [241]. Error estimates for interpolation using Gaussian processes are developed in [332]. The link between regression and Bayesian inversion with Gaussian process priors is developed in depth in [66, 328]. A definition of MAP estimator in infinite dimensions, appropriate for the discussion in Remark 14.11 may be found in [72]. We have articulated a specific link between neural networks and random features; a related concept underpins the subject of the *neural tangent kernel* [162]. The use of various machine-learning inspired approximation methods in the solution of PDEs may be found in [262, 56, 230, 175].

# Chapter 15

## Time Series Forecasting

In most other chapters we have used $n$ to index data and $j$ to index (discrete) time in (possibly stochastic) dynamical systems. In this chapter the two are conflated as the data comes from time series:

**Data Assumption 15.1.** *Let* $\mathsf{N} = \{0, 1, \cdots, N-1\}$. *We are given a time-series* $\{v_n^\dagger\}_{n\in\mathsf{N}}$, *of elements in* $\mathbb{R}^d$.

For this reason the (possibly stochastic) dynamical systems in this chapter will be indexed by $n$ not $j$. The following formal and informal definitions will be useful to us:

**Definition 15.2.** A process $\{v_n\}$ is *strictly stationary* if, for all $p$, the joint distribution of $(v_n, \dots, v_{n+p})$ does not change with $n$. ♢

We use the following informal definition of an ergodic time series. A process $\{v_n\}$, in $\mathbb{R}^d$, is *ergodic* if there is probability measure $\mu$ on $\mathbb{R}^d$ such that, for a sufficiently wide class of test functions $\varphi$,

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \varphi(v_n) = \int_{\mathbb{R}^d} \varphi(v)\, \mu(dv). \tag{15.1}$$

If the process is deterministic it is possible that measure $\mu$ will not have density with respect to Lebesgue measure which is why, in contrast to much of the remainder of the notes, we have not used a probability density function in this definition.

In Section 15.1 we introduce linear autoregressive models. Section 15.2 studies analog forecasting methods. In Section 15.3 we look at fitting Markovian models to the data, and non-Markovian models with a recurrent structure to capture memory. Section 15.4 introduces non-Gaussian autoregressive models, generalizing the content of Section 15.1. We conclude in Section 15.6 with bibliographic remarks.

## 15.1 Linear Autoregressive Models

A vector autoregressive (VAR) process is one of the most simple time-series models.

**Definition 15.3.** The $p$th-order VAR model, written VAR($p$), is

$$v_n = c + \sum_{\ell=1}^{p} A_\ell\, v_{n-\ell} + \varepsilon_n, \tag{15.2}$$

where the $A_\ell$ are constant matrices, $c$ is a constant vector, and $\varepsilon_n \sim \mathcal{N}(0, C)$. When $v_n$ is a scalar, the process is known simply as an autoregressive (AR) model. Note that $\varepsilon_n$ can also be a different strictly stationary and ergodic process, but we use Gaussian errors here for simplicity. ♢

**Theorem 15.4.** *If $\{v_n\}$ is a stationary VAR(1) process with $c = 0$, then the distribution of $v_n$ is $\mathcal{N}(0, C_\infty)$, where*

$$\mathrm{vec}(C_\infty) = (I - A \otimes A)^{-1}\,\mathrm{vec}(C).$$

*Note that we take $c = 0$ for simplicity, but the proof is easily modified for $c \neq 0$.*

*Proof.* First note that under Gaussian errors, the marginal distribution of $v_n$ is Gaussian (see the Bibliography section for references). By stationarity, we must have that

$$\mathbb{E}v_n = A_1 \mathbb{E}v_{n-1} = \mathbb{E}v_{n-1},$$

which implies that $\mathbb{E}v_n = 0$. We must also have that

$$\mathbb{E}[v_n v_n^\top] = A_1 \mathbb{E}[v_{n-1} v_{n-1}^\top] A_1^\top + C = \mathbb{E}[v_{n-1} v_{n-1}^\top],$$

where the second equality follows from elementary properties of Gaussians. This is a discrete Lyapunov equation, which can be solved using properties of vectorization. We obtain

$$\mathrm{vec}(\mathbb{E}[v_{n-1} v_{n-1}^\top]) = (I - A_1 \otimes A_1)^{-1}\,\mathrm{vec}(C).$$

□

## 15.2 Analog Forecasting

We again make Data Assumption 15.1. The goal of *forecasting* is to predict $v_q$ given $v_0$, and given the time-series data provided in Data Assumption 15.1. *Analog forecasting* specifically operates by predicting from given specific point, by looking for nearby points in data and seeing how they evolved.

### 15.2.1 Analog Forecasting

Let $\mathsf{D}$ be a distance-like deterministic scoring rule on $\mathbb{R}^d$ from Definition 12.56, recalling that any norm on $\mathbb{R}^d$ provides an example. *Analog forecasting* works as follows. Let

$$n^\star \in \arg\min_{n \in \mathsf{N}} \mathsf{D}(v_0, v_n^\dagger). \tag{15.3}$$

Then the desired prediction is, for $n^\star < N - q$,

$$v_q = v_{n^\star + q}^\dagger;$$

for $n^\star \geq N - q$ no prediction is made because the data does not support doing so.

The intuition behind the forecast is simple to articulate: find the closest point in the data set to $v_0$ and ask what happened $q$ time units later in the data set; this is used as the prediction.

Remark 15.5. The predictions produced by analog forecasting are discontinuous with respect to changes in $v_0$, since the closest point to $v_0$ can change as it is varied continuously. In the next subsection we describe a generalization which removes this discontinuous behaviour. ♢

Remark 15.6. The success of analog forecasting will depend on the availability of close analogs to $v_0$, as well as the Lyapunov spectrum of the system that generated $\{v_n^\dagger\}$. The first point is intuitive. We now explain the second point. Suppose that $\{v_n^\dagger\}$ is generated by the deterministic dynamical system $\Psi$, so that

$$v_{n+1}^\dagger = \Psi(v_n^\dagger).$$

Suppose further that $\Psi$ has a global attractor, supporting an invariant measure $\mu$. Let $\Psi^{(n)}$ denote the $n$-fold composition of $\Psi$ with itself. Under conditions described by Oseledet's Theorem (see Bibliography Section 15.6), we can define the maximal *Lyapunov exponent* of $\Psi$ as

$$\lambda_{\max} = \lim_{n\to\infty} \lim_{\epsilon\to 0^+} \frac{1}{n} \log \frac{|\Psi^{(n)}(v^\dagger + \epsilon e) - \Psi^{(n)}(v^\dagger)|}{\epsilon}, \tag{15.4}$$

for $\mu$- almost every $v^\dagger$ and almost every vector $e$ picked uniformly at random from the unit sphere.

For simplicity consider the case $\Psi(v) = Av$ where $A$ is positive and symmetric (and hence diagonalizable) and the spectral radius of $A$, $\lambda$, is attained at unique eigenvector $\varphi$. This does not satisfy the standard assumptions of the theory, but is simple enough to provide intuition. Provided that the projection of $e$ onto $\varphi$ is non-zero (which will happen almost surely) it is clear that $\lambda_{\max} = \lambda$. Thus the maximal Lyapunov exponent delivers the typical growth rate of perturbations to the system.

We now discuss how Lyapunov exponents manifest in analog forecasting. Recall that the analog forecast from $v_0$ is found by identifying $n$ such that $v_n^\dagger$ is the closest point in the data set to $v_0$ and setting $v_q = v_{n+q}^\dagger = \Psi^{(q)}(v_n^\dagger)$. Thus, approximately for $|v_n^\dagger - v_0|$ small, we have that on average

$$|v_q - \Psi^{(q)}(v_0)| \approx |v_n^\dagger - v_0| e^{\lambda_{\max} q}. \tag{15.5}$$

♢

### 15.2.2 Kernel Analog Forecasting

Let $p : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ be a positive-definite *kernel*. Then make prediction

$$v_q = \frac{1}{N-q} \sum_{n=0}^{N-1-q} p(v_0, v_n^\dagger) v_{n+q}^\dagger.$$

This method, too, has a simple intuitive explanation, generalizing analog forecasting. Rather than forecasting using $q$ steps ahead from a single closest point in the data set to the initial condition, $q$-step forecasts are made from all points in the data set (for which it is possible to forecast $q$ steps ahead). These are then weighted according to how close the starting point in the data set is to $v_0$; the kernel performs this weighting. A typical choice of kernel is the radial basis function kernel $p(v,w)=\exp(-|v-w|^2/\lambda)$, and the choice of $\lambda$ is crucial to the success of the algorithm.

Remark 15.7. Assuming that the kernel is continuous in its arguments it follows that this forecast will be continuous with respect to $v_0$. $\diamondsuit$

## 15.3 Recurrent Structure

We again seek to forecast but now we attempt to do so by learning a map from the data given in Data Assumption 15.1.

### 15.3.1 Markovian Prediction

We seek to explain the data $\{v_n^\dagger\}_{n=0}^{N-1}$ as being the outcome of deterministic Markovian map on $\mathbb{R}^d$, of form

$$v_{n+1}=\psi(v_n;\theta).$$

We can view this as a generalization of the supervised learning problem, defined by Data Assumption 14.1, where the inputs are no longer i.i.d. The function $\psi(\cdot;\theta)$ can be, for example, a neural network, a random features model or the mean of a Gaussian process as in Chapter 14.

To learn $\theta$ define

$$\mathsf{J}^N(\theta)=\frac{1}{N}\sum_{n=0}^{N-1}\bigl|v_{n+1}^\dagger-\psi(v_n^\dagger;\theta)\bigr|^2, \tag{15.7a}$$

$$\theta^\star\in\arg\min_{\theta\in\Theta}\mathsf{J}^N(\theta). \tag{15.7b}$$

The population-level optimization problem that this approximates may be defined as follows. Assume that data is given in the form of probability density function $\pi(v,w)$ in $\mathcal{P}(\mathbb{R}^d\times\mathbb{R}^d)$, representing the two-point correlation function of a stationary ergodic process. Then define

$$\mathsf{J}(\theta)=\mathbb{E}^{(v,w)\sim\pi}\bigl|w-\psi(v;\theta)\bigr|^2, \tag{15.8a}$$

$$\theta^\star\in\arg\min_{\theta\in\Theta}\mathsf{J}(\theta). \tag{15.8b}$$

If data $\{v_n^\dagger\}_{n=0}^{N-1}$ is derived from such a stationary ergodic process, then $\mathsf{J}(\theta)$ is the pointwise limit of $\mathsf{J}^N(\theta)$, by ergodicity.

### 15.3.2 Memory and Prediction

The premise of the preceding subsection is that data $\{v_n^\dagger\}_{n=0}^{N-1}$ derives from a Markov process and there often arise settings where this is not the case. In such a setting it is of interest to introduce an unobserved latent variable $\{r_n\}_{n=0}^{N-1}$, with $r_n\in\mathbb{R}^r$ and then hypothesize a model of the form

$$v_{n+1}=\psi(v_n,r_n;\theta_v), \tag{15.9a}$$

$$r_{n+1}=\varphi(v_n,r_n;\theta_r). \tag{15.9b}$$

If $\psi, \varphi$ are neural networks then the overall structure is known as a *recurrent neural network*, or RNN for short. We define $\theta = (\theta_v, \theta_r) \in \Theta = \Theta_v \times \Theta_v$ and learn $\theta$ by solving the minimization problem

$$\mathsf{J}^N(\theta) = \frac{1}{N} \sum_{n=0}^{N-1} \big|v_{n+1}^{\dagger} - \psi(v_n^{\dagger}, r_n; \theta_v)\big|^2, \tag{15.10a}$$

$$r_{n+1} = \varphi(v_n^{\dagger}, r_n; \theta_r),\ r_0 = 0, \tag{15.10b}$$

$$\theta^\star \in \arg\min_{\theta \in \Theta} \mathsf{J}^N(\theta). \tag{15.10c}$$

Remark 15.8. The choice $r_0 = 0$ is arbitrary and a change of initial condition can always be shifted to the origin by redefining the right-hand side. However it is also possible to learn the best choice of $r_0$, using data assimilation, given training data in form of an observed time-series in the space where sequence $\{v_n\}$ lives. $\diamondsuit$

Remark 15.9. The need for memory in time-series prediction can be motivated not only for non-Markovian systems, but also for Markovian but partially observed systems. One way to see this is using *Takens' Theorem*, which says (roughly) that under quite general conditions, given partial observations of a Markovian system but enough memory, one can reconstruct the original dynamics.

We state Takens' Theorem informally; see the bibliography for references. Suppose that $\Psi : \mathbb{R}^d \to \mathbb{R}^d$ defines a deterministic dynamical system, $v_{j+1}^{\dagger} = \Psi(v^{\dagger})$, with an invariant set $A$. Then, define a *time-delay vector* of embedding dimension $m$ for a scalar-valued observation function $h : \mathbb{R}^d \to \mathbb{R}$,

$$H(v_j^{\dagger}) = \Big(h(v_{j-m}^{\dagger}), \ldots, h(v_{j-1}^{\dagger})\Big).$$

This function $H$ is defined everywhere on $A$. For $m$ large enough and quite generic choices of $h$, $A \to H(A)$ is one-to-one. $\diamondsuit$

### 15.3.3 Memory and Reservoir Computing

Including memory is important in many time series prediction tasks as the data may not come from a Markov process. However learning RNN structure can be challenging for two reasons: (i) because $r_n$ is an unobserved latent variable; (ii) and because, within the context of gradient-based methods (Chapter 16) propagation of gradients of $\theta_r$ through the model is difficult. Reservoir computing is one approach to overcome this. Rather than learning $\theta_r$ it is simply picked at random from some probability measure on space of appropriate dimension and the following optimization problem is solved:

$$\mathsf{J}^N(\theta_v) = \frac{1}{N} \sum_{n=0}^{N-1} \big|v_{n+1}^{\dagger} - \psi(v_n^{\dagger}, r_n; \theta_v)\big|^2, \tag{15.11a}$$

$$r_{n+1} = \varphi(v_n^{\dagger}, r_n; \theta_r),\ r_0 = 0, \tag{15.11b}$$

$$\theta^\star \in \arg\min_{\theta_v \in \Theta_v} \mathsf{J}^N(\theta_v). \tag{15.11c}$$

Crucially the optimization is only over $\theta_v$ as $\theta_r$ is fixed at a randomly chosen value.

Remark 15.10. Typically the method requires $r$ of higher dimension than for the RNN in order to be successful. This is related to the issue of how expressive randomly chosen functions are. Indeed there is a connection to the random features methodology of Section 14.2. To see this, note that equation (15.11b) reveals that

$$r_n = \mathcal{F}_n(\{v_m^\dagger\}_{m=0}^{n-1}; \theta_r).$$

Thus, once $\theta_r$ is chosen at random, $r_n$ is a random function of the history of the observed data up to time $n-1$.

♢

## 15.4 Non-Gaussian Auto-Regressive Models

In this section we extend the autoregressive models with additive Gaussian noise in Section 15.1 to have a general probabilistic form. One approach to model non-Gaussian distributions uses transport (see Chapter 5) to represent the state as a transformation of a Gaussian variable. That is, we can seek a transport $T \in \mathbb{R}^{(n+1)d} \times \mathbb{R}^p$ depending on parameters $\theta \in \mathbb{R}^p$ that satisfies

$$T(v_1, \dots, v_n, \epsilon; \theta) \sim \mathbb{P}(v_{n+1}|v_1, \dots, v_n), \quad \epsilon \sim \mathcal{N}(0, I), \tag{15.12}$$

for each $(v_1, \dots, v_n) \in \mathbb{R}^{nd}$. Let $\pi(v_{n+1}|v_1, \dots, v_n)$ denote the conditional density of the next step $v_{n+1}$ and let $\varrho(v)$ be the standard Gaussian density, both on $\mathbb{R}^d$. Then, the transport in (15.12) satisfies the pushforward condition $T(v_1, \dots, v_n, \cdot)_\sharp \varrho(v_{n+1}) = \pi(v_{n+1}|v_1, \dots, v_n)$.

If $T$ is invertible with respect to $\epsilon$, we can use the change-of-variables formula to express the density of the transformed variable as

$$\pi(v_{n+1}|v_1, \dots, v_n) = \varrho(T(v_1, \dots, v_n, \cdot; \theta)^{-1}|_{v_{n+1}}) \det D_{v_{n+1}} T(v_1, \dots, v_n, \cdot; \theta)^{-1}|_{v_{n+1}}.$$

**Example 15.11.** If $T(v_1, \dots, v_n, \epsilon; \theta) = \Psi(v_n; \theta) + C^{1/2}\epsilon$, for some map $\Psi \colon \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ as in Subsections 15.3.1 or 15.3.2 and a positive definite matrix $C \in \mathbb{R}^{d \times d}$ we have

$$\begin{aligned}\pi(v_{n+1}|v_1, \dots, v_n) &= \varrho\Big(C^{-1/2}(v_{n+1} - \Psi(v_n))\Big) \det C^{-1/2} \\ &= \frac{1}{\sqrt{(2\pi)^d |C|}} \exp\Big(-|v_{n+1} - \Psi(v_n)|_C^2\Big).\end{aligned}$$

That is, the conditional distribution for the next state $\pi(v_{n+1}|v_1, \dots, v_n) = \pi(v_{n+1}|v_n)$ is a multivariate Gaussian of the form $\mathcal{N}(\Psi(v_n), C)$. ♢

These transports permit one to learn general autoregressive models that depend on an arbitrary number of past states. In this section we will learn transports that depend on at most one past state variable under Data Assumption 15.1. In this case, our goal is to find invertible transports $T \colon \mathbb{R}^{2d} \times \mathbb{R}^p \to \mathbb{R}^d$ so that $\pi(v_{n+1}|v_n, \theta) \coloneqq T(v_n, \epsilon; \theta)_\sharp \varrho(v_{n+1})$ models the conditional distribution for the state at the next time. To simplify the notation, we denote the inverse map $T(v_n, \cdot)^{-1}|_{v_{n+1}}$ by $S(v_n, v_{n+1}; \theta)$.

Given a sequence of data, we can learn the transport parameters by maximizing the log-likelihood of consecutive data pairs $\{(v_{n+1}^\dagger, v_n^\dagger)\}_{n=0}^{N-2}$ under the model. For a standard-Gaussian reference $\varrho$, the log-likelihood is given by the loss function

$$\begin{aligned}
\mathsf{L}(\theta) &= \frac{1}{N-1}\sum_{n=0}^{N-1} \log \pi(v_{n+1}^\dagger | v_n^\dagger, \theta) \\
&= \frac{1}{N-1}\sum_{n=0}^{N-1} \Big[\log \varrho(S(v_n^\dagger, v_{n+1}^\dagger;\theta)) + \log\det D_{v_{n+1}} S(v_n^\dagger, v_{n+1}^\dagger;\theta)\Big] \\
&= \frac{1}{N-1}\sum_{n=0}^{N-1} \Big[-\frac{1}{2}|S(v_n^\dagger, v_{n+1}^\dagger)|^2 + \log\det D_{v_{n+1}} S(v_n^\dagger, v_{n+1}^\dagger)\Big] + c, \qquad (15.13)
\end{aligned}$$

where $c$ is a constant that is independent of $\theta$. The optimal transport parameters are then given by

$$\theta^\star \in \arg\min_\theta \mathsf{L}(\theta).$$

**Example 15.12.** One popular non-Gaussian transformation is based on a nonlinear diagonal transformation of the form

$$S(v_n, v_{n+1};\theta) = s(v_{n+1} - \Psi(v_n;\theta);\theta),$$

where $s \in \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ is an invertible function and $\Psi \in \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$. This can be re-written in terms of the direct transport as

$$T(v_n, \epsilon;\theta) = \Psi(v_n;\theta) + s^{-1}(\epsilon;\theta), \quad \epsilon \sim \varrho.$$

That is, the next-state is given by the previous state with additive noise, which is non-Gaussian for nonlinear $s$. The joint distribution for the states at all times $n$ that follow this process is often referred to as a *non-paranormal* or *Gaussian copula model*. ♢

**Example 15.13.** If $S(v_n, v_{n+1};\theta) = v_{n+1} - \Psi(v_n;\theta)$, as in example (15.11) with $C = I$, we have $D_{v_{n+1}} S(v_n, v_{n+1}) = I$. Thus, the learning problem based on the loss function in (15.13) reduces to

$$\theta^\star \in \arg\min_\theta \frac{1}{2(N-1)}\sum_{n=0}^{N-1}\Big[|v_{n+1}^\dagger - \Psi(v_n^\dagger;\theta)|^2\Big],$$

That is, we can identify the model parameters by solving a supervised learning problem for the nonlinear map $\Psi$. ♢

## 15.5 Singular Spectrum Analysis

*Singular spectrum analysis* (SSA) decomposes a time-series into spatiotemporal modes $\{r^{(k)}\}_k$, which can then be forecasted using the other methods described in this chapter. These modes can be chosen to correspond to physically relevant or predictable components of the time series, and modes corresponding to noise discarded. Thus, SSA can be used as a preprocessing step prior to forecasting.

SSA applies principal component analysis to time-delay embedded data. Given an embedding window $L$, which should be chosen based on the timescales of interest, we proceed to form the *trajectory matrix*

$$X = \begin{pmatrix} v_0^\dagger & v_1^\dagger & \cdots & v_{N-L+1}^\dagger \\ \vdots & \vdots & \ddots & \vdots \\ v_{L-1}^\dagger & v_L^\dagger & \cdots & v_N^\dagger \end{pmatrix}^\top .$$

Then, performing principal component analysis on $X$, we eigendecompose the covariance matrix $C = X^\top X/(N-L+1)$ obtaining a set of eigenvalues $\{\lambda^{(k)}\}$ and eigenvectors $\{e^{(k)}\}$. The matrix $X$ can then be projected onto the eigenvector $e^{(k)}$:

$$a^{(k)} = Xe^{(k)}.$$

The $k$th *reconstructed component* $r^{(k)}$, the portion of $\{v_n^\dagger\}$ that corresponds to mode $e^{(k)}$, is formed by convolving $a^{(k)}$ with $e^{(k)}$:

$$r_n^{(k)} = \frac{1}{U_n - L_n + 1} \sum_{m=L_n}^{U_n} a_{n-m+1}^{(k)} e_m^{(k)},$$

where

$$(L_n, U_n) = \begin{cases} (0, n-1) & \text{if } 0 \le n \le L-2, \\ (0, L-1) & \text{if } L-1 \le n \le N-L+1, \\ (n-N+L-1, L-1) & \text{if } N-L+2 \le n \le N. \end{cases}$$

Summing all the $\{r^{(k)}\}$ returns the original time series $\{v_n^\dagger\}$. Summing over a subset of the $\{r^{(k)}\}$ corresponds to a partial reconstruction of the time series, which may be more favorable for forecasting.

## 15.6 Bibliography

An extensive introduction to linear autoregressive processes, including the conditions for stationarity and ergodicity, is provided in [130]. The Lorenz '63 model is proved to be ergodic in [313, 314] providing a useful example. Indeed in this context central limit theorems have been proved [148], characterizing the fluctuations about the law of large numbers limit in (15.1).

Analog forecasting was first introduced by Lorenz in [202]. A historical overview of analog forecasting and extensions is given in [94]. The kernel analog regularization of Lorenz's idea was introduced and then studied in [111, 5].

Memory in the context of learning model error is discussed in [194]. Takens' Theorem and other embedding theorems are reviewed in [281].

The idea behind SSA was first introduced in [42], and further developed in [319]. It is reviewed in detail in [110]. Aspects related to forecasting with SSA are discussed in [237]. Nonlinear generalizations are considered in [112].

General non-linear autoregressive models have been considered for time-series modeling using various probabilistic models, such as conditional normalizing flows, in [238, 266]. Some of these models rely on recurrent neural networks architectures that can summarize the past states and extract relevant features for making predictions.

# Chapter 16

## Optimization

We provide here an introduction to various topics in optimization that are pertinent to these notes. Section 16.1 is concerned with gradient-based optimization, including deterministic and stochastic gradient descent. In Section 16.2 we study auto-differentiation. Section 16.3 is devoted to expectation-maximization. In Section 16.4 we discuss Newton and Gauss-Newton methods. Section 16.5 is devoted to ensemble Kalman methods for parameter estimation and inversion.[1] Other topics in optimization are referenced in the bibliography Section 16.6.

Before moving into these sections we recall a definition that will be useful in what follows.

**Definition 16.1.** A function $f : \mathbb{R}^d \to \mathbb{R}$ is called *convex* if, for all $x, y \in \mathbb{R}^d$ and for all $\theta \in [0, 1]$, we have that

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta)f(y).$$

A function is called *strictly convex* if the inequality is strict for $x \neq y$. $\diamondsuit$

## 16.1 Gradient Descent

### 16.1.1 Deterministic Gradient Descent

**Definition 16.2.** Given function $f \in C^1(\mathbb{R}^d, \mathbb{R})$, *gradient descent* refers to the iterative generation of sequence $\{x_j\}_{j\in\mathbb{Z}^+}$ from a sequence of step-sizes $\{\alpha_j\}_{j\in\mathbb{Z}^+}$ by picking an $x_0 \in \mathbb{R}^d$ and then iterating as follows:

$$x_{j+1} = x_j - \alpha_j Df(x_j), \qquad j \in \mathbb{Z}^+.$$

$\diamondsuit$

Remark 16.3. Often the sequence $\{\alpha_j\}_{j\in\mathbb{Z}^+}$ is determined online as the algorithm progresses, with $\alpha_j$ depending on $\{x_i\}_{i=1}^j$. In the simplest case, $\alpha_j = \alpha$ is some fixed constant, chosen small enough that $f(x_j)$ decreases with increasing $j$; however being

[1] Although this topic might be argued to belong in the first part of the book (Inverse Problems), it builds on ensemble Kalman methods from the second part of the book (Data Assimilation) which is why we cover it in this appendix chapter from part three of the book.

able to make such a choice depends on having bounds on the size of $Df$. The next theorem considers a case where such bounds follow from assumptions about the Hessian of $f$. $\diamondsuit$

The following theorem demonstrates that a specific fixed $\alpha$, under assumptions on the Hessian, leads to a linear convergence of gradient descent to the global minimum.

**Theorem 16.4.** *Let $f \in C^2(\mathbb{R}^d, \mathbb{R})$ with Hessian satisfying, for all $x \in \mathbb{R}^d$,*

$$\mu I \preceq D^2 f(x) \preceq LI$$

*for some $0 < \mu \leq L < \infty$. Then $f(\cdot)$ has a unique global minimum $x^\star$. Furthermore, at iteration $j \geq 1$ of gradient descent initialized at $x_0$ with step size $\alpha_j \equiv 1/L$,*

$$f(x_j) - f(x^\star) \leq \left(1 - \frac{\mu}{L}\right)^j \bigl(f(x_0) - f(x^\star)\bigr).$$

*Proof.* The existence of a unique global minimizer is a consequence of the convexity of $f$ implied by the bounds on the Hessian. By Taylor's theorem for a twice-differentiable function $f$ we have

$$\begin{aligned} f(y) &= f(x) + \langle Df(x), y - x\rangle + \frac{1}{2}(y-x)^\top H(y-x), \\ H &= \int_0^1 D^2 f(sx + (1-s)y)\, ds. \end{aligned}$$

From the uniform bound on the Hessian, we have the upper and lower bounds

$$f(y) \leq f(x) + \langle Df(x), y - x\rangle + \frac{L}{2}|y-x|^2, \tag{16.1a}$$

$$f(y) \geq f(x) + \langle Df(x), y - x\rangle + \frac{\mu}{2}|y-x|^2. \tag{16.1b}$$

Taking $y = x_{j+1}$ and $x = x_j$ in (16.1a) with the gradient descent update $x_{j+1} = x_j - \alpha_j Df(x_j)$ we have

$$\begin{aligned} f(x_{j+1}) &\leq f(x_j) + \langle Df(x_j), x_{j+1} - x_j\rangle + \frac{L}{2}|x_{j+1} - x_j|^2 \\ &= f(x_j) - \alpha_j |Df(x_j)|^2 + \frac{L}{2}|\alpha_j Df(x_j)|^2 \\ &= f(x_j) - \frac{1}{2L}|Df(x_j)|^2, \end{aligned} \tag{16.2}$$

where in the last line we have used that $\alpha_j = 1/L$.

Now take $y = x^\star$ and $x = x_j$ in (16.1b) and multiply both sides by a negative sign to get

$$f(x_j) - f(x^\star) \leq \langle Df(x_j), x_j - x^\star\rangle - \frac{\mu}{2}|x^\star - x_j|^2.$$

Adding and subtracting the norm of the gradient to the right-hand-side, we have

$$\begin{aligned} f(x_j) - f(x^\star) &\le \langle Df(x_j), x_j - x^\star\rangle - \frac{\mu}{2}|x^\star - x_j|^2 \\ &= \frac{1}{2\mu}|Df(x_j)|^2 - \frac{1}{2\mu}|Df(x_j)|^2 + \langle Df(x_j), x_j - x^\star\rangle - \frac{\mu}{2}|x^\star - x_j|^2 \\ &= \frac{1}{2\mu}|Df(x_j)|^2 - \frac{1}{2}\left|\sqrt{\frac{1}{\mu}}Df(x_j) - \sqrt{\mu}(x^\star - x_j)\right|^2 \\ &\le \frac{1}{2\mu}|Df(x_j)|^2. \end{aligned}$$

Re-arranging the terms we have $-|Df(x_j)|^2 \le -2\mu\big(f(x_j) - f(x^\star)\big)^2$. Substituting this on the right-hand side of (16.2) gives us

$$f(x_{j+1}) \le f(x_j) - \frac{\mu}{L}\big(f(x_j) - f(x^\star)\big).$$

If we subtract $f(x^\star)$ from both sides and let $e_{j+1} = f(x_{j+1}) - f(x^\star)$, we have

$$e_{j+1} \le e_j - \frac{\mu}{L}e_j = \left(1 - \frac{\mu}{L}\right)e_j$$

for all $j \ge 1$. By induction on $j$, we have the desired result. □

Remark 16.5. Consider convex functions with Lipschitz gradients, properties that follow from the Hessian bounds in the preceding theorem. For such objective functions there are modifications of gradient descent that have better convergence rates than the one stated above. For instance, Nesterov's accelerated gradient descent has a quadratic convergence rate, which is optimal for this class of functions. We will not discuss these other methods here but provide references in the bibliography. ♢

### 16.1.2 Stochastic Gradient Descent

Here we consider the minimization of an objective function $f : \mathbb{R}^d \to \mathbb{R}$ that is defined by an expectation of a function $F \colon \mathbb{R}^d \times \mathbb{R}^{d_z} \to \mathbb{R}$. That is,

$$f(x) = \int F(x,z)\, d\zeta(z). \tag{16.3}$$

Here $\zeta$ denotes the probability density function of the random variable $z \in \mathbb{R}^{d_z}$. While explicit evaluations of $f$ or its gradients are challenging as a result of the expectation, we assume that it is possible to evaluate $D_x F(x,z)$ for any realization of the random variable $z$.

Assuming that $\zeta$ is replaced by an empirical approximation

$$\zeta^N = \frac{1}{N}\sum_{n=1}^{N} \delta_{z^{(n)}},$$

with the $z^{(n)}$ sampled i.i.d. from $\zeta$, then $f$ becomes

$$f(x) = \frac{1}{N}\sum_{n=1}^{N} F(x, z^{(n)}). \tag{16.4}$$

The gradient of $f$ is then given by

$$Df(x) = \frac{1}{N}\sum_{n=1}^{N} D_x F(x, z^{(n)}). \tag{16.5}$$

This also has the interpretation as a Monte Carlo estimator of the gradient of $f$ defined by (16.3).

For large $N$, the evaluation of the above sum may be too expensive. Instead, one can pick $M < N$ i.i.d. samples from $\zeta^N$; this corresponds to picking $M$ random indices $\{\omega^{(m)}\}_{m=1}^{M}$, where the elements of this set $\omega^{(m)}$ are sampled uniformly with replacement from $1, \ldots, N$. Then the gradient can be approximated as

$$Df(x) \approx \frac{1}{M}\sum_{m=1}^{M} D_x F(x, z^{(\omega^{(m)})}).$$

This is known as a *mini-batch*. Carrying out gradient descent with this mini-batch is known as *stochastic gradient descent*.

**Definition 16.6.** Given function $F \in C^{1,0}(\mathbb{R}^d \times \mathbb{R}^{d_z}, \mathbb{R})$, *stochastic gradient descent* with respect to $f$ defined by (16.4), refers to the iterative generation of sequence $\{x_j\}_{j\in\mathbb{Z}^+}$ from a sequence of step-sizes $\{\alpha_j\}_{j\in\mathbb{Z}^+}$ and $j$-indexed sets $\{\omega_j^{(m)}\}_{m=1}^{M}$ with points sampled uniformly with replacement from $1, \ldots, N$, independently for each $j$, by picking an $x_0 \in \mathbb{R}^d$ and then iterating as follows:

$$x_{j+1} = x_j - \frac{\alpha_j}{M}\sum_{m=1}^{M} D_x F(x_j, z^{(\omega_j^{(m)})}).$$

$\diamondsuit$

Remark 16.7. Even if the full gradient with respect to the empirical measure (16.5) can be feasibly computed, the stochasticity in stochastic gradient descent can be helpful in escaping from local minima for non-convex objective functions; in the context of optimizing neural networks it is also sometimes advocated as a method to avoid overfitting. $\diamondsuit$

## 16.2 Automatic Differentiation

Gradients (and higher-order derivatives) are often difficult to obtain in closed form for complex cost functions. Finite difference approximations, on the other hand, are often inaccurate and expensive for high-dimensional problems.

*Automatic differentiation* involves repeated application of chain rule on elementary operations that enables computing derivatives accurately to working precision. That is, suppose an output variable $y \in \mathbb{R}^{d_n}$ is related to an input variable $x \in \mathbb{R}^{d_0}$ by some function $f$, i.e., $y = f(x)$, and suppose that the Jacobian of $f$ is not readily available in closed form. We assume the implementation of $f$ in computer code is made up of $n$ elementary operations $f_i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$ (for instance, addition, multiplication, logarithms, etc.), such that

$$f(x) = f_n \circ f_{n-1} \circ \cdots \circ f_1(x), \tag{16.6}$$

where the Jacobians for these elementary operations, $Df_i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i \times d_{i-1}}$, are available in closed form. The goal of automatic differentiation is to evaluate $Df(x)$ from the Jacobians of the elementary operations.

By representing the output $g_i = f_i(g_{i-1})$ of each elementary operations by a node, we can use a graph to represent the function evaluation where the directed edges encode the input dependencies $g_{i-1}$ (i.e., parents) that are required to evaluate $f_i$. This is known as the *computational graph*. The nodes without parents correspond to the original inputs $x$ to the function $f$. After constructing the graph, we can define the *forward mode* as the process of computing the output $y$, and possibly the gradient of $f$, by traversing the graph starting from the inputs $x$. The *reverse mode* starts with the final output node and traverses the graph back to the inputs to compute the gradient of $f$. The following two subsections demonstrate how the full Jacobian of $f$ is assembled in forward and reverse mode automatic differentiation.

Remark 16.8. In realistic implementations, instead of multiplying the full Jacobians of each operation, only the partial derivatives of variables that interact with each other in the computational graph are used. For simplicity of exposition, we present the algorithms with the full Jacobians here. ♢

### 16.2.1 Forward Mode

Forward mode automatic differentiation proceeds to accumulate the incremental function evaluations along with their derivatives.

Writing the partial evaluations up to $i \le n$ as

$$g_i = (f_i \circ \cdots \circ f_1)(x),$$

by the chain rule we have that

$$D_x f(x) = \big(D_{g_{n-1}} f_n(g_{n-1})\big) \cdots \big(D_{g_1} f_2(g_1)\big)\big(D_x f_1(x)\big). \tag{16.7}$$

This suggests the following algorithm:

**Algorithm 16.1** Forward Mode Automatic Differentiation

```
Input: The functions $\{f_i(\cdot)\}_{i=1}^n$, their corresponding Jacobians $\{Df_i(\cdot)\}_{i=1}^n$, and the function input $x$.
Set $g_1 = f_1(x)$ and $J_1 = Df_1(x)$.
For $i = 2, \ldots, n$: set $g_i = f_i(g_{i-1})$ and $J_i = \big(Df_i(g_{i-1})\big)J_{i-1}$.
Output: The function output $y = f(x) = g_n$ and the derivative $Df(x) = J_n$.
```

The number of floating point operations required to compute $Df(x)$ is

$$d_0 \sum_{i=1}^{n-1} d_{i+1} d_i.$$

Assuming for simplicity that all the $d_i$ except for $d_0$ and $d_n$ are fixed at $d$, we have a cost of $d^2(n-2)d_0 + d_n d d_0$. This scales favorably for large $d_n$ (large output dimension), but not for large $d_0$ (large input dimension).

### 16.2.2 Reverse Mode

Motivated by the poor scaling of forward mode automatic differentiation with input dimension, *reverse mode* automatic differentiation also computes the product of Jacobians (16.7),

$$D_x f(x) = \big(D_{g_{n-1}} f_n(g_{n-1})\big) \cdots \big(D_{g_1} f_2(g_1)\big)\big(D_x f_1(x)\big),$$

but instead of computing the product from right to left, it computes it from left to right. The algorithmic implementation requires a forward pass to evaluate the function and a backwards pass to compute the derivative:

**Algorithm 16.2** Reverse Mode Automatic Differentiation

```
Input: The functions $\{f_i(\cdot)\}_{i=1}^n$, their corresponding Jacobians $\{Df_i(\cdot)\}_{i=1}^n$, and the function input $x$.
Set $g_1 = f_1(x)$.
For $i = 2, \ldots, n$: set and store $g_i = f_i(g_{i-1})$.
Set $J_n = Df_n(g_{n-1})$.
For $i = n-1, \ldots, 1$ set $J_i = J_{i+1} Df_i(g_{i-1})$.
Output: The function output $y = f(x) = g_n$ and the derivative $Df(x) = J_1$.
```

This leads to the number of floating point operations in computing the Jacobian being

$$d_n \sum_{i=1}^{n-1} d_{i-1} d_i.$$

Again assuming that all the $d_i$ are fixed at $d$ except for $d_0$ and $d_n$, the total cost is $d^2(n-2)d_n + d_n d d_0$. As opposed to forward mode automatic differentiation, reverse mode thus scales favorably with a large number of inputs (large $d_0$), but poorly with a large number of outputs (large $d_n$).

Remark 16.9. Backpropagation, a common method for computing the gradient of the objective function with respect to the weights of a neural network, is a special case of reverse mode differentiation. ♢

## 16.3 Expectation-Maximization

The expectation-maximization (EM) algorithm is a general-purpose approach to maximum likelihood estimation with latent variables. Central to the abstract derivation of the EM algorithm is the following lower bound of the log-likelihood function, obtained using Jensen inequality:

$$\begin{aligned}\log \mathbb{P}(Y|\theta) &= \log \int \mathbb{P}(V, Y|\theta)\, dV \\ &= \log \int \frac{\mathbb{P}(V, Y|\theta)}{q(V)} q(V)\, dV \\ &\geq \int \log\Bigl(\frac{\mathbb{P}(V, Y|\theta)}{q(V)}\Bigr) q(V)\, dV \\ &= \mathsf{ELBO}(\mathbb{P}(V, Y|\theta), q) \\ &=: \mathcal{L}(q, \theta),\end{aligned}$$

where $q$ is any probability density function over the latent variables $V$ with compatible support, ELBO denotes the evidence lower bound introduced in Definition 2.13, and the last line defines $\mathcal{L}(q, \theta)$. In fact, the following theorem shows that the difference between $\log \mathbb{P}(Y|\theta)$ and $\mathcal{L}(q, \theta)$ is given by the KL divergence $\mathsf{D}_{\mathrm{KL}}\bigl(q \| \,\mathbb{P}(V|Y, \theta)\bigr)$, which is always non-negative. Notice that the proof is identical to that of the variational formulation of Bayes theorem in Chapter 2. A similar result was also used in the proof of Theorem 9.3 in Chapter 9.

**Theorem 16.10.** *It holds that*

$$\log \mathbb{P}(Y|\theta) = \mathcal{L}(q, \theta) + \mathsf{D}_{\mathrm{KL}}\bigl(q \| \,\mathbb{P}(V|Y, \theta)\bigr).$$

*Proof.* Using the definition of $\mathcal{L}(q, \theta)$, product rule, and the definition of the KL divergence in Chapter 12, we have

$$\begin{aligned}\mathcal{L}(q, \theta) &= \int \log\Bigl(\frac{\mathbb{P}(V, Y)|\theta)}{q(V)}\Bigr) q(V)\, dV \\ &= \int \log\Bigl(\frac{\mathbb{P}(V|Y, \theta)}{q(V)}\Bigr) q(V)\, dV + \int \log \mathbb{P}(Y|\theta) q(V)\, dV \\ &= -\mathsf{D}_{\mathrm{KL}}\bigl(q \| \,\mathbb{P}(V|Y, \theta)\bigr) + \log \mathbb{P}(Y|\theta),\end{aligned}$$

as desired. □

Theorem 16.10 motivates an iterative approach to maximize the log-likelihood function $\log \mathbb{P}(Y|\theta)$, which is equivalent to maximizing the likelihood $\mathbb{P}(Y|\theta)$ since the logarithm is an increasing function. Given the current iterate $\theta_j$, we obtain the next iterate $\theta_{j+1}$ in two steps, maximizing in turn the two components of the lower bound:

1. First, we find $q_j(V)$ that maximizes the lower bound $\mathcal{L}(q, \theta_j)$ over probability density function $q$. From Theorem 16.10,
$$\log \mathbb{P}(Y|\theta_j) = \mathcal{L}(q, \theta_j) + \mathsf{D}_{\mathrm{KL}}\big(q \| \, \mathbb{P}(V|Y, \theta_j)\big),$$
and it follows that maximizing $\mathcal{L}(q, \theta_j)$ or minimizing $\mathsf{D}_{\mathrm{KL}}\big(q \| \, \mathbb{P}(V|Y, \theta_j)\big)$ over $q$ are equivalent. Since the KL divergence is minimized when both arguments agree, we obtain that $q_j(V) = \mathbb{P}(V|Y, \theta_j)$.
2. Second, we obtain $\theta_{j+1}$ by maximizing the lower bound $\mathcal{L}(q_j, \theta)$ over $\theta$. Note that
$$\mathcal{L}(q_j, \theta) = \int \log \mathbb{P}(V, Y|\theta) \, \mathbb{P}(V|Y, \theta_j) \, dV + c, \tag{16.8}$$
where $c$ is independent of $\theta$. Hence, the quantity to maximize is the expected value of the joint log-density $\log \mathbb{P}(V, Y|\theta)$ with respect to $q_j(V) = \mathbb{P}(V|Y, \theta_j)$.

Combining these two steps gives the EM framework:

**Algorithm 16.3** Expectation Maximization

```
1: Input: Initialization $\theta_0$.
2: For $j = 0, 1, \dots$ do the following expectation and maximization steps:
3: E-Step: Compute
       $$\mathbb{E}^{V \sim \mathbb{P}(V|Y,\theta_j)} \Big[\log \mathbb{P}(V, Y|\theta)\Big] = \int \log \mathbb{P}(V, Y|\theta) \, \mathbb{P}(V|Y, \theta_j) \, dV.$$
4: M-Step: Compute
       $$\theta_{j+1} \in \arg\max_{\theta} \mathbb{E}^{V \sim \mathbb{P}(V|Y,\theta_j)} \Big[\log \mathbb{P}(V, Y|\theta)\Big].$$
5: Output: Parameter estimates $\{\theta_j\}_{j \ge 0}$.
```

The following result shows that the EM framework has the desirable property that the likelihood function increases monotonically along iterates $\theta_j$.

**Theorem 16.11.** *Let $\{\theta_j\}_{j\ge 0}$ be the iterates of the EM Algorithm 16.3. Then,*
$$\log \mathbb{P}(Y|\theta_j) \le \log \mathbb{P}(Y|\theta_{j+1}). \tag{16.9}$$

*Proof.* Let $q_j(V) = \mathbb{P}(V|Y, \theta_j)$. Using the log-likelihood characterization in Theorem 16.10, it holds that
$$\begin{aligned} \log \mathbb{P}(Y|\theta_{j+1}) &= \mathcal{L}(q_j, \theta_{j+1}) + \mathsf{D}_{\mathrm{KL}}\big(q_j \| \, \mathbb{P}(V|Y, \theta_{j+1})\big) \\ &\ge \mathcal{L}(q_j, \theta_j) + \mathsf{D}_{\mathrm{KL}}\big(q_j \| \, \mathbb{P}(V|Y, \theta_{j+1})\big) \\ &\ge \mathcal{L}(q_j, \theta_j) + \mathsf{D}_{\mathrm{KL}}\big(q_j \| \, \mathbb{P}(V|Y, \theta_j)\big) \\ &= \log \mathbb{P}(Y|\theta_j). \end{aligned}$$
The first inequality follows because $\theta_{j+1} \in \arg\max_\theta \mathcal{L}(q_j, \theta)$; the second because $\mathsf{D}_{\mathrm{KL}}\big(q_j \| \, \mathbb{P}(V|Y, \theta_j)\big) = 0$ since $q_j = \mathbb{P}(V|Y, \theta_j)$, while $\mathsf{D}_{\mathrm{KL}}\big(q_j \| \, \mathbb{P}(V|Y, \theta_{j+1})\big) \ge 0$. □

Remark 16.12. As a consequence of Theorem 16.11 it is possible to deduce—under mild assumptions—that the iterates $\theta_j$ of the EM algorithm converge, as $j \to \infty$, to a local maximizer of the likelihood function. It is important to note, however, that the expectation in the E-step and the optimization in the M-step are often intractable. Monte Carlo, filtering, or smoothing algorithms may be employed to approximate the E-step, and optimization algorithms to approximate the M-step. Such approximations can cause loss of monotonicity and convergence guarantees. $\diamondsuit$

## 16.4 Newton and Gauss–Newton

### 16.4.1 Newton's Method

Newton's method is a common root-finding method, which can also be applied to optimization.

**Definition 16.13.** Given function $f \in C^2(\mathbb{R}^d, \mathbb{R})$, with invertible Hessian, *Newton's method* refers to the iterative generation of sequence $\{x_j\}_{j\in\mathbb{Z}^+}$ by picking an $x_0 \in \mathbb{R}^d$ and then iterating as follows:

$$x_{j+1} = x_j - \big(D^2 f(x_j)\big)^{-1} D f(x_j). \tag{16.10}$$

$\diamondsuit$

Newton's method proceeds by optimizing successive quadratic approximations to $f$, as the following theorem shows.

**Theorem 16.14.** *If $f$ is convex, then $f(x_{j+1})$ is the local minimum of the quadratic approximation of $f$ around $x_j$.*

*Proof.* We first approximate $f$ around $x_j$ by its second-order Taylor expansion:

$$f(x_j + t) \approx f(x_j) + \langle Df(x_j), t\rangle + \frac{1}{2} t^\top D^2 f(x_j) t, \tag{16.11}$$

where $t$ is some vector. Since $f$ is convex, the Hessian is positive semidefinite, and the right-hand side of (16.11) is convex as a function of $t$, meaning that it can be minimized by setting the derivative to 0. Differentiating $f(x_j + t)$ with respect to $t$ and setting it to 0,

$$Df(x_j) + D^2 f(x_j) t = 0,$$

and thus

$$t = -\big(D^2 f(x_j)\big)^{-1} Df(x_j).$$

$\square$

### 16.4.2 Gauss–Newton

Let $g$ be an $\mathbb{R}^m$-valued function and consider the nonlinear least squares optimization problem

$$\mathsf{J}(x) = \frac{1}{2} g(x)^\top g(x) = \frac{1}{2}\sum_{k=1}^{m} g_k(x)^2, \tag{16.12a}$$

$$x^\star \in \arg\min_{x\in\mathbb{R}^d} \mathsf{J}(x). \tag{16.12b}$$

**Definition 16.15.** Given function $g \in C^1(\mathbb{R}^d, \mathbb{R}^m)$ such that $\big((Dg(x_j)\big)^\top (Dg(x_j))$ is invertible, the *Gauss–Newton method* for minimizing nonlinear least squares problems of the form (16.12) refers to the iterative generation of sequence $\{x_j\}_{j\in\mathbb{Z}^+}$ by picking an $x_0 \in \mathbb{R}^d$ and then iterating as follows:

$$x_{j+1} = x_j - \Big(\big(Dg(x_j)\big)^\top \big(Dg(x_j)\big)\Big)^{-1} \big(Dg(x_j)\big)^\top g(x_j).$$

$\diamondsuit$

Gauss–Newton results from applying Newton's method (16.10) to (16.12a), and neglecting the second-order terms of $g$ in the Hessian of $\mathsf{J}$. Notice that

$$D\mathsf{J}(x) = \big(Dg(x)\big)^\top g(x),$$

and that

$$\big(D^2\mathsf{J}(x)\big)_{ij} = \big(Dg(x)^\top Dg(x)\big)_{ij} + \sum_{k=1}^{m} g_k(x)\big(D^2 g_k(x)\big)_{ij}. \tag{16.13}$$

Substituting these expressions into Newton's method (16.10) and setting the second term in (16.13) to 0, recovers the Gauss–Newton method.

## 16.5 Ensemble Kalman Inversion

We describe here a derivative-free optimization method for nonlinear least squares problems. Consider the cost function

$$\mathsf{J}(u) = \frac{1}{2}|y - G(u)|_\Gamma^2,$$

for some (possibly) nonlinear function $G(\cdot)$ and positive definite matrix $\Gamma$.

*Ensemble Kalman inversion* (EKI) involves first drawing an initial set of $N$ particles from a Gaussian distribution with a given mean $m_0$ and covariance $C_0$, i.e.,

$$u_0^{(n)} \sim \mathcal{N}(m_0, C_0) \quad \text{i.i.d.}, \quad n = 1, \dots, N.$$

EKI proceeds to evolve these particles for iterations $j = 1, \dots, J$ according to

$$u_{j+1}^{(n)} = u_j^{(n)} + K_{j+1}\big(y - \eta_{j+1}^{(n)} - G(u_j^{(n)})\big), \quad n = 1, \dots, N, \tag{16.14a}$$

$$\eta_{j+1}^{(n)} \sim \mathcal{N}(0, \Gamma) \quad \text{i.i.d.}, \tag{16.14b}$$

where the matrix $K_{j+1}$ is calculated according to

$$\overline{u}_{j+1} = \frac{1}{N}\sum_{n=1}^{N} u_{j+1}^{(n)}, \tag{16.15a}$$

$$\overline{G}_{j+1} = \frac{1}{N}\sum_{n=1}^{N} G(u_{j+1}^{(n)}), \tag{16.15b}$$

$$C_{j+1}^{ug} = \frac{1}{N}\sum_{n=1}^{N} (u_{j+1}^{(n)} - \overline{u}_{j+1}) \otimes (G(u_{j+1}^{(n)}) - \overline{G}_{j+1}), \tag{16.15c}$$

$$C_{j+1}^{gg} = \frac{1}{N}\sum_{n=1}^{N} (G(u_{j+1}^{(n)}) - \overline{G}_{j+1}) \otimes (G(u_{j+1}^{(n)}) - \overline{G}_{j+1}), \tag{16.15d}$$

$$C_{j+1}^{yy} = C_{j+1}^{gg} + \Gamma, \quad K_{j+1} = C_{j+1}^{ug}(C_{j+1}^{yy})^{-1}. \tag{16.15e}$$

Remark 16.16. The algorithm produces an ensemble $\{u_j^{(n)}\}_{n=1}^N$ which, for each $j$, remains in the linear span of the initial ensemble $\{u_0^{(n)}\}_{n=1}^N$. Thus the algorithm may be viewed as seeking to minimize the objective function over a finite-dimensional subspace. In the case of linear $G(\cdot)$ this leads to a thorough theoretical understanding of the algorithm, and variants on it including Tikhonov regularization and continuous time. Furthermore, EKI can often still approximate the minimizer of the loss function well in the nonlinear setting, with a relatively small $J$ and $N$. ♢

Remark 16.17. As the name suggests, there is similarity between EKI and the ensemble Kalman filter (EnKF: Section 7.6). At every iteration, EKI approximates the solution to a nonlinear inverse problem using an ensemble approximation to the solution of a linear Gaussian inverse problem (Example 1.8) and formally replacing the linear forward model with the nonlinear one. ♢

## 16.6 Bibliography

Optimization is reviewed in a number of textbooks, including [38, 235]. Convergence proofs for gradient descent and stochastic gradient descent may be found in [278] in simple settings. Nesterov's accelerated gradient method and its rate of convergence is discussed in detail in [231]. Gradient descent can be formulated in continuous time as a *gradient flow*; see [126, 6]. A detailed overview of automatic differentiation is given in [122].

The EM framework for maximum likelihood estimation was introduced in [78]; see [218] for a review and see [274] for an insightful connection to Gibbs sampling. The generalization of the EM framework for optimization problems where the objective does not necessarily involve conditional expectations is known as the Minorize-Maximization (MM), or equivalently Majorize-Minimization, algorithm. An overview of MM algorithms and their properties can be found in [182].

The book [235] includes discussion of Newton and Gauss–Newton methods.

Ensemble Kalman inversion (EKI) as defined here was introduced in [160], and various adaptations of it may be found in [157, 158, 51, 159]; it is also discussed in the final chapter of the textbook [278]. For analysis in continuous time see [284, 285]. A mean field perspective on EKI and an interpretation of the method as a covariance-preconditioned gradient flow is provided in [45]. Variants of EKI are discussed in [153, 152, 154].

# Bibliography

[1] S. Agapiou, M. Burger, M. Dashti, and T. Helin. Sparsity-promoting and edge-preserving maximum a posteriori estimators in non-parametric Bayesian inverse problems. *Inverse Problems*, 34(4):045002, 2017.

[2] S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3): 405–431, 2017.

[3] O. Al-Ghattas and D. Sanz-Alonso. Non-asymptotic analysis of ensemble Kalman updates: Effective dimension and localization. *Information and Inference: A Journal of the IMA*, 13(1):iaad043, Mar. 2024. ISSN 2049-8772. doi: 10.1093/imaiai/iaad043.

[4] M. Al-Jarrah, N. Jin, B. Hosseini, and A. Taghvaei. Optimal transport-based nonlinear filtering in high-dimensional settings. *arXiv preprint arXiv:2310.13886*, 2023.

[5] R. Alexander and D. Giannakis. Operator-theoretic framework for forecasting nonlinear time series with kernel analog techniques. *Physica D: Nonlinear Phenomena*, 409:132520, 2020.

[6] L. Ambrosio, N. Gigli, and G. Savaré. *Gradient Flows: in Metric Spaces and in the Space of Probability Measures.* Springer Science & Business Media, 2005.

[7] J. Amezcua and P. J. van Leeuwen. Time-correlated model error in the (ensemble) Kalman smoother. *Quarterly Journal of the Royal Meteorological Society*, 144 (717):2650–2665, 2018. ISSN 1477-870X. doi: 10.1002/qj.3378.

[8] J. Anderson. Spatially and temporally varying adaptive covariance inflation for ensemble filters. *Tellus A: Dynamic Meteorology and Oceanography*, 61(1):72–83, Jan. 2009. ISSN null. doi: 10.1111/j.1600-0870.2007.00361.x.

[9] J. L. Anderson. An ensemble adjustment Kalman filter for data assimilation. *Monthly Weather Review*, 129(12):2884–2903, 2001.

[10] J. L. Anderson. A non-Gaussian ensemble filter update for data assimilation. *Monthly Weather Review*, 138(11):4186–4198, 2010.

[11] C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient monte carlo computations. 2009.

[12] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[13] C. Archambeau, M. Opper, Y. Shen, D. Cornford, and J. Shawe-Taylor. Variational inference for diffusion processes. *Advances in Neural Information Processing Systems*, 20, 2007.

[14] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

[15] M. Asch, M. Bocquet, and M. Nodet. *Data Assimilation: Methods, Algorithms, and Applications*, volume 11. SIAM, 2016.

[16] M. Asim, M. Daniels, O. Leong, A. Ahmed, and P. Hand. Invertible generative models for inverse problems: mitigating representation error and dataset bias. In *International Conference on Machine Learning*, pages 399–409. PMLR, 2020.

[17] E. Bach and M. Ghil. A multi-model ensemble Kalman filter for data assimilation and forecasting. *Journal of Advances in Modeling Earth Systems*, 15(1): e2022MS003123, Jan. 2023. ISSN 1942-2466. doi: 10.1029/2022MS003123.

[18] E. Bach, S. Mote, V. Krishnamurthy, A. S. Sharma, M. Ghil, and E. Kalnay. Ensemble Oscillation Correction (EnOC): Leveraging Oscillatory Modes to Improve Forecasts of Chaotic Systems. *Journal of Climate*, 34(14):5673–5686, July 2021. ISSN 0894-8755, 1520-0442. doi: 10.1175/JCLI-D-20-0624.1.

[19] A. Bain and D. Crisan. *Fundamentals of Stochastic Filtering*, volume 60. Springer Science & Business Media, 2008.

[20] G. Bal. Introduction to Inverse Problems. *Lecture Notes-Department of Applied Physics and Applied Mathematics, Columbia University, New York*, 2012.

[21] R. Baptista, B. Hosseini, N. B. Kovachki, and Y. Marzouk. Conditional sampling with monotone GANs: from generative models to likelihood-free inference. *arXiv e-prints*, pages arXiv–2006, 2020.

[22] R. Baptista, B. Hosseini, N. B. Kovachki, Y. M. Marzouk, and A. Sagiv. An approximation theory framework for measure-transport sampling algorithms. *arXiv preprint arXiv:2302.13965*, 2023.

[23] R. Baptista, Y. Marzouk, and O. Zahm. On the representation and learning of monotone triangular transport maps. *Foundations of Computational Mathematics*, pages 1–46, 2023.

[24] D. Basu. On the elimination of nuisance parameters. *Journal of the American Statistical Association*, 72(358):355–366, 1977. ISSN 01621459. URL http://www.jstor.org/stable/2286800.

[25] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, and C. Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021.

[26] M. Benning and M. Burger. Modern regularization methods for inverse problems. *Acta Numerica*, 27:1–111, 2018.

[27] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer Science & Business Media, 2011.

[28] P. Bickel, B. Li, and T. Bengtsson. Sharp failure rates for the bootstrap particle filter in high dimensions. In *Pushing the limits of contemporary statistics: Contributions in honor of Jayanta K. Ghosh*, pages 318–329. Institute of Mathematical Statistics, 2008.

[29] D. Bigoni, Y. Chen, N. Garcia Trillos, Y. Marzouk, and D. Sanz-Alonso. Data-driven forward discretizations for Bayesian inversion. *Inverse Problems*, 36(10): 105008, 2020.

[30] C. M. Bishop. *Pattern Recognition and Machine Learning*, volume 128. Springer, 2006.

[31] A. Block, Y. Mroueh, and A. Rakhlin. Generative modeling with denoising auto-encoders and Langevin sampling. *arXiv preprint arXiv:2002.00107*, 2020.

[32] M. Bocquet. Surrogate modeling for the climate sciences dynamics with machine learning and data assimilation. *Frontiers in Applied Mathematics and Statistics*, 9, 2023. ISSN 2297-4687. doi: 10.3389/fams.2023.1133226.

[33] M. Bocquet, J. Brajard, A. Carrassi, and L. Bertino. Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization. *Foundations of Data Science*, 2(1):55–80, 2020.

[34] T. Bolton and L. Zanna. Applications of deep learning to ocean data inference and subgrid parameterization. *Journal of Advances in Modeling Earth Systems*, 11(1):376–399, 2019.

[35] M. Bonavita. On the limitations of data-driven weather forecasting models. *arXiv preprint arXiv:2309.08473*, 2023.

[36] M. Bonavita and P. Laloyaux. Machine learning for model error inference and correction. *Journal of Advances in Modeling Earth Systems*, 12(12):e2020MS002232, 2020.

[37] P. Boudier, A. Fillion, S. Gratton, S. Gürol, and S. Zhang. Data assimilation networks. *Journal of Advances in Modeling Earth Systems*, 15(4):e2022MS003353, 2023.

[38] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, 2004. ISBN 978-0-521-83378-3. doi: 10.1017/CBO9780511804441.

[39] J. Brajard, A. Carrassi, M. Bocquet, and L. Bertino. Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the lorenz 96 model. *Journal of Computational Science*, 44:101171, 2020.

[40] J. Bröcker and I. G. Szendro. Sensitivity and out-of-sample error in continuous time data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 138 (664):785–801, 2012. ISSN 1477-870X. doi: 10.1002/qj.940.

[41] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov chain Monte Carlo.* CRC press, 2011.

[42] D. S. Broomhead and G. P. King. Extracting qualitative dynamics from experimental data. *Physica D: Nonlinear Phenomena*, 20(2):217–236, June 1986. ISSN 0167-2789. doi: 10.1016/0167-2789(86)90031-X.

[43] T. A. Brown. Admissible scoring systems for continuous distributions. The Rand Paper Series P-5235, RAND Corporation, Santa Monica, CA, 1974.

[44] S. L. Brunton and J. N. Kutz. *Data-driven Science and Engineering: Machine Learning, Dynamical Systems, and Control.* Cambridge University Press, 2019.

[45] E. Calvello, S. Reich, and A. M. Stuart. Ensemble Kalman Methods: A Mean Field Perspective. *arXiv*, 2022.

[46] A. Carrassi and S. Vannitsem. Deterministic Treatment of Model Error in Geophysical Data Assimilation. In F. Ancona, P. Cannarsa, C. Jones, and A. Portaluri, editors, *Mathematical Paradigms of Climate Science*, pages 175–213. Springer International Publishing, Cham, 2016. ISBN 978-3-319-39092-5. doi: 10.1007/978-3-319-39092-5_9.

[47] A. Carrassi, M. Bocquet, A. Hannart, and M. Ghil. Estimating model evidence using data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(703):866–880, 2017.

[48] A. Carrassi, M. Bocquet, L. Bertino, and G. Evensen. Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *Wiley Interdisciplinary Reviews: Climate Change*, 9(5), 2018.

[49] A. Carrassi, M. Bocquet, J. Demaeyer, C. Grudzien, P. Raanes, and S. Vannitsem. Data assimilation for chaotic dynamics. In S. K. Park and L. Xu, editors, *Data*

*Assimilation for Atmospheric, Oceanic and Hydrologic Applications (Vol. IV)*, pages 1–42. Springer International Publishing, Cham, 2022. ISBN 978-3-030-77722-7. doi: 10.1007/978-3-030-77722-7_1.

[50] J. Carrillo, F. Hoffmann, A. Stuart, and U. Vaes. The ensemble Kalman filter in the near-Gaussian setting. *arXiv preprint arXiv:2212.13239*, 2022.

[51] N. K. Chada, A. M. Stuart, and X. T. Tong. Tikhonov regularization within ensemble Kalman inversion. *SIAM Journal on Numerical Analysis*, 58(2):1263–1294, 2020.

[52] A. Chattopadhyay, M. Mustafa, P. Hassanzadeh, E. Bach, and K. Kashinath. Towards physics-inspired data-driven weather forecasting: integrating data assimilation with a deep spatial-transformer-based U-NET in a case study with ERA5. *Geoscientific Model Development*, 15(5):2221–2237, 2022.

[53] A. Chattopadhyay, E. Nabizadeh, E. Bach, and P. Hassanzadeh. Deep learning-enhanced ensemble-based data assimilation for high-dimensional nonlinear dynamical systems. *Journal of Computational Physics*, 477:111918, Mar. 2023. ISSN 0021-9991. doi: 10.1016/j.jcp.2023.111918.

[54] N. Chen. *Stochastic Methods for Modeling and Predicting Complex Dynamical Systems: Uncertainty Quantification, State Estimation, and Reduced-Order Models.* Synthesis Lectures on Mathematics & Statistics. Springer International Publishing, Cham, 2023. ISBN 978-3-031-22248-1 978-3-031-22249-8. doi: 10.1007/978-3-031-22249-8.

[55] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018.

[56] Y. Chen, B. Hosseini, H. Owhadi, and A. M. Stuart. Solving and learning nonlinear PDEs with Gaussian processes. *Journal of Computational Physics*, 447:110668, 2021.

[57] Y. Chen, D. Sanz-Alonso, and R. Willett. Auto-differentiable ensemble Kalman filters. *SIAM Journal on Mathematics of Data Science*, 4(2):801–833, 2022.

[58] Y. Chen, D. Z. Huang, J. Huang, S. Reich, and A. M. Stuart. Gradient flows for sampling: Mean-field models, Gaussian approximations and affine invariance. *arXiv preprint arXiv:2302.11024*, 2023.

[59] S. Cheng, J.-P. Argaud, B. Iooss, A. Ponçot, and D. Lucor. A Graph Clustering Approach to Localization for Adaptive Covariance Tuning in Data Assimilation Based on State-Observation Mapping. *Mathematical Geosciences*, 53(8):1751–1780, Nov. 2021. ISSN 1874-8953. doi: 10.1007/s11004-021-09951-z.

[60] S. Cheng, C. Quilodrán-Casas, S. Ouala, A. Farchi, C. Liu, P. Tandeo, R. Fablet, D. Lucor, B. Iooss, J. Brajard, D. Xiao, T. Janjic, W. Ding, Y. Guo, A. Carrassi, M. Bocquet, and R. Arcucci. Machine learning with data assimilation and uncertainty quantification for dynamical systems: a review. *IEEE/CAA Journal of Automatica Sinica*, 10(6):1361–1387, June 2023. ISSN 2329-9274. doi: 10.1109/JAS.2023.123537.

[61] H. G. Chipilski. Exact nonlinear state estimation. *arXiv preprint arXiv:2310.10976*, 2023.

[62] E. Cleary, A. Garbuno-Inigo, S. Lan, T. Schneider, and A. M. Stuart. Calibrate, emulate, sample. *Journal of Computational Physics*, 424:109716, 2021.

[63] T. J. Cocucci, M. Pulido, M. Lucini, and P. Tandeo. Model error covariance estimation in particle and ensemble Kalman filters using an online expectation–maximization algorithm. *Quarterly Journal of the Royal Meteorological Society*, 147(734):526–543, 2021.

[64] S. Cotter, M. Dashti, and A. Stuart. Variational data assimilation using targetted random walks. *International Journal for Numerical Methods in Fluids*, 68(4): 403–421, 2012.

[65] K. Cranmer, J. Brehmer, and G. Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

[66] P. Craven and G. Wahba. Smoothing noisy data with spline functions. *Numerische mathematik*, 31(4):377–403, 1978.

[67] D. Crisan and B. Rozovskii. *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011.

[68] M. Croci, K. E. Willcox, and S. J. Wright. Multi-output multilevel best linear unbiased estimators via semidefinite programming. *Computer Methods in Applied Mechanics and Engineering*, 413:116130, Aug. 2023. ISSN 0045-7825. doi: 10.1016/j.cma.2023.116130.

[69] L. Csato, D. Cornford, and M. Opper. Data assimilation with sequential Gaussian processes. *Uncertainty in Geometric Computations*, pages 29–39, 2002.

[70] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[71] M. Dashti and A. M. Stuart. Bayesian approach to inverse problems. *Handbook of Uncertainty Quantification*, pages 311–428, 2017.

[72] M. Dashti, K. J. H. Law, A. M. Stuart, and J. Voss. MAP estimators and their consistency in Bayesian nonparametric inverse problems. *Inverse Problems*, 29(9): 095017, 2013.

[73] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 20(4): 633–679, 2020.

[74] D. P. Dee and A. M. Da Silva. Data assimilation in the presence of forecast bias. *Quarterly Journal of the Royal Meteorological Society*, 124(545):269–295, 1998. ISSN 1477-870X. doi: 10.1002/qj.49712454512.

[75] P. Del Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer Science & Business Media, 2004.

[76] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[77] T. DelSole and X. Yang. State and parameter estimation in stochastic dynamical models. *Physica D: Nonlinear Phenomena*, 239(18):1781–1788, 2010.

[78] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[79] M. Destouches, P. Mycek, and S. Gürol. Multivariate extensions of the Multilevel Best Linear Unbiased Estimator for ensemble-variational data assimilation, June 2023.

[80] R. DeVore, B. Hanin, and G. Petrova. Neural network approximation. *Acta Numerica*, 30:327–444, 2021.

[81] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[82] A. Doucet, N. d. Freitas, and N. Gordon. An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

[83] D. Dreano, P. Tandeo, M. Pulido, B. Ait-El-Fquih, T. Chonavel, and I. Hoteit. Estimating model-error covariances in nonlinear state-space models using Kalman smoothing and the expectation–maximization algorithm. *Quarterly Journal of the Royal Meteorological Society*, 143(705):1877–1885, 2017.

[84] C. Drovandi, R. G. Everitt, A. Golightly, D. Prangle, et al. Ensemble MCMC: accelerating pseudo-marginal MCMC for state space models using the ensemble Kalman filter. *Bayesian Analysis*, 2021.

[85] O. R. Dunbar, A. Garbuno-Inigo, T. Schneider, and A. M. Stuart. Calibration and uncertainty quantification of convective parameters in an idealized GCM. *Journal of Advances in Modeling Earth Systems*, 13(9):e2020MS002454, 2021.

[86] M. M. Dunlop, D. Slepčev, A. M. Stuart, and M. Thorpe. Large data and zero noise limits of graph-based semi-supervised learning algorithms. *Applied and Computational Harmonic Analysis*, 49(2):655–697, 2020.

[87] W. E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 1(5):1–11, 2017.

[88] T. A. El Moselhy and Y. M. Marzouk. Bayesian inference with optimal maps. *Journal of Computational Physics*, 231(23):7815–7850, 2012.

[89] H. England, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Springer Science and Business Media, 1996.

[90] G. Evensen. 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, 1995.

[91] G. Evensen, F. C. Vossepoel, and P. J. van Leeuwen. *Data Assimilation Fundamentals: A Unified Formulation of the State and Parameter Estimation Problem*. Springer, 2022.

[92] R. Fakoor, T. Kim, J. Mueller, A. J. Smola, and R. J. Tibshirani. Flexible model aggregation for quantile regression. *Journal of Machine Learning Research*, 24 (162):1–45, 2023.

[93] A. Farchi, P. Laloyaux, M. Bonavita, and M. Bocquet. Using machine learning to correct model error in data assimilation and forecast applications. *Quarterly Journal of the Royal Meteorological Society*, 147(739):3067–3084, 2021.

[94] J. D. Farmer and J. J. Sidorowich. Exploiting Chaos to Predict the Future and Reduce Noise. In Y. C. Lee, editor, *Evolution, Learning and Cognition*, pages 277–330. World Scientific, Singapore, 1988. ISBN 978-9971-5-0529-5. doi: 10.1142/9789814434102_0011.

[95] C. A. T. Ferro. Measuring forecast performance in the presence of observation error. *Quarterly Journal of the Royal Meteorological Society*, 143(708):2665–2676, 2017. ISSN 1477-870X. doi: 10.1002/qj.3115.

[96] C.-N. Fiechter. PAC adaptive control of linear systems. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, pages 72–80, 1997.

[97] A. Filoche, J. Brajard, A. Charantonis, and D. Béréziat. Learning 4DVAR inversion directly from observations. In *International Conference on Computational Science*, pages 414–421. Springer, 2023.

[98] M. Fisher and E. Andersson. Developments in 4D-Var and Kalman filtering. 2001.

[99] M. Fraccaro, S. Kamronn, U. Paquet, and O. Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. *arXiv preprint arXiv:1710.05741*, 2017.

[100] T. E. Fricker, C. A. T. Ferro, and D. B. Stephenson. Three recommendations for evaluating climate predictions. *Meteorological Applications*, 20(2):246–255, 2013. ISSN 1469-8080. doi: 10.1002/met.1409.

[101] M. Gabrié, G. M. Rotskoff, and E. Vanden-Eijnden. Efficient Bayesian sampling using normalizing flows to assist Markov chain Monte Carlo methods. *arXiv preprint arXiv:2107.08001*, 2021.

[102] T. Galy-Fajou, V. Perrone, and M. Opper. Flexible and efficient inference with particles for the variational Gaussian approximation. *Entropy*, 23(8):990, 2021.

[103] A. F. Gao, O. Leong, H. Sun, and K. L. Bouman. Image reconstruction without explicit priors. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[104] A. Garbuno-Inigo, T. Helin, F. Hoffmann, and B. Hosseini. Bayesian posterior perturbation analysis with integral probability metrics. *arXiv preprint arXiv:2303.01512*, 2023.

[105] N. Garcia Trillos and D. Sanz-Alonso. Continuum limits of posteriors in graph Bayesian inverse problems. *SIAM Journal on Mathematical Analysis*, 50(4): 4020–4040, 2018.

[106] N. Garcia Trillos and D. Sanz-Alonso. The Bayesian update: variational formulations and gradient flows. *Bayesian Analysis*, 15(1):29–56, 2020.

[107] N. Garcia Trillos, Z. Kaplan, T. Samakhoana, and D. Sanz-Alonso. On the consistency of graph-based Bayesian semi-supervised learning and the scalability of sampling algorithms. *Journal of Machine Learning Research*, 21(28):1–47, 2020.

[108] A. Genevay, G. Peyré, and M. Cuturi. Learning generative models with Sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617. PMLR, 2018.

[109] M. Ghil, S. Cohn, J. Tavantzis, K. Bube, and E. Isaacson. Applications of Estimation Theory to Numerical Weather Prediction. In L. Bengtsson, M. Ghil, and E. Källén, editors, *Dynamic Meteorology: Data Assimilation Methods*, Applied Mathematical Sciences, pages 139–224. Springer, New York, NY, 1981. ISBN 978-1-4612-5970-1. doi: 10.1007/978-1-4612-5970-1_5.

[110] M. Ghil, M. R. Allen, M. D. Dettinger, K. Ide, D. Kondrashov, M. E. Mann, A. W. Robertson, A. Saunders, Y. Tian, F. Varadi, and P. Yiou. Advanced Spectral Methods for Climatic Time Series. *Reviews of Geophysics*, 40(1):3–1–3–41, Feb. 2002. ISSN 1944-9208. doi: 10.1029/2000RG000092.

[111] D. Giannakis. Data-driven spectral decomposition and forecasting of ergodic dynamical systems. *Applied and Computational Harmonic Analysis*, 47(2):338–396, 2019.

[112] D. Giannakis and A. J. Majda. Nonlinear Laplacian spectral analysis: Capturing intermittent and low-frequency spatiotemporal patterns in high-dimensional data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 6(3): 180–194, 2013. ISSN 1932-1872. doi: 10.1002/sam.11171.

[113] A. Gibbs and F. Su. On choosing and bounding probability metrics. *International Statistical Review*, 70(3):419–435, 2002.

[114] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[115] T. Gneiting and R. Ranjan. Comparing density forecasts using threshold- and quantile-weighted scoring rules. *Journal of Business & Economic Statistics*, July 2011. doi: 10.1198/jbes.2010.08110.

[116] T. Gneiting, L. I. Stanberry, E. P. Grimit, L. Held, and N. A. Johnson. Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds. *TEST*, 17(2):211–235, Aug. 2008. ISSN 1863-8260. doi: 10.1007/s11749-008-0114-x.

[117] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[118] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT press, 2016.

[119] G. A. Gottwald and S. Reich. Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation. *Physica D: Nonlinear Phenomena*, 423:132911, 2021.

[120] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

[121] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[122] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Society for Industrial and Applied Mathematics, 2 edition, 2008. ISBN 978-0-89871-659-7. doi: 10.1137/1.9780898717761.

[123] I. Grooms. A comparison of nonlinear extensions to the ensemble Kalman filter: Gaussian anamorphosis and two-step ensemble filters. *Computational Geosciences*, 26(3):633–650, 2022.

[124] A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions.* Chapman and Hall/CRC, 2018.

[125] E. Haber and L. Ruthotto. Stable architectures for deep neural networks. *Inverse problems*, 34(1):014004, 2017.

[126] J. K. Hale. *Asymptotic behavior of dissipative systems.* Number 25. American Mathematical Soc., 2010.

[127] F. Hamilton, T. Berry, and T. Sauer. Ensemble Kalman filtering without a model. *Physical Review X*, 6(1):011021, Mar. 2016. doi: 10.1103/PhysRevX.6.011021.

[128] M. A. E. R. Hammoud, N. Raboudi, E. S. Titi, O. Knio, and I. Hoteit. Data assimilation in chaotic systems using deep reinforcement learning. *Journal of Advances in Modeling Earth Systems*, 16(8):e2023MS004178, 2024.

[129] A. Hannart, A. Carrassi, M. Bocquet, M. Ghil, P. Naveau, M. Pulido, J. Ruiz, and P. Tandeo. Dada: data assimilation for the detection and attribution of weather and climate-related events. *Climatic Change*, 136(2):155–174, 2016.

[130] B. E. Hansen. *Econometrics.* Princeton University Press, 2022. ISBN 978-0-691-23589-9.

[131] J. Harlim, D. Sanz-Alonso, and R. Yang. Kernel methods for Bayesian elliptic inverse problems on manifolds. *SIAM/ASA Journal on Uncertainty Quantification*, 8(4):1414–1445, 2020.

[132] J. Harlim, S. W. Jiang, S. Liang, and H. Yang. Machine learning for prediction with missing dynamics. *Journal of Computational Physics*, 428:109922, 2021.

[133] J. Harlim, S. W. Jiang, H. Kim, and D. Sanz-Alonso. Graph-based prior and forward models for inverse problems on manifolds with boundaries. *Inverse Problems*, 38(3):035006, 2022.

[134] T. Harris, B. Li, and R. Sriver. Multimodel ensemble analysis with neural network Gaussian processes. *The Annals of Applied Statistics*, 17(4):3403–3425, Dec. 2023. ISSN 1932-6157, 1941-7330. doi: 10.1214/23-AOAS1768.

[135] F. P. Härter and H. Fraga de Campos Velho. Data assimilation procedure by recurrent neural network. *Engineering Applications of Computational Fluid Mechanics*, 6(2):224–233, Jan. 2012. ISSN 1994-2060. doi: 10.1080/19942060.2012.11015417.

[136] S. Hatfield, M. Chantry, P. Dueben, P. Lopez, A. Geer, and T. Palmer. Building Tangent-Linear and Adjoint Models for Data Assimilation With Neural Networks. *Journal of Advances in Modeling Earth Systems*, 13(9):e2021MS002521, 2021. ISSN 1942-2466. doi: 10.1029/2021MS002521.

[137] T. Helin and M. Burger. Maximum a posteriori probability estimates in infinite-dimensional Bayesian inverse problems. *Inverse Problems*, 31(8):085009, 2015.

[138] T. Helin, A. Stuart, A. Teckentrup, and K. Zygalakis. Introduction to Gaussian process regression in Bayesian inverse problems, with new results on experimental design for weighted error measures. *arXiv preprint arXiv:2302.04518*, 2023.

[139] J. Hernandez-Lobato, Y. Li, M. Rowland, T. Bui, D. Hernández-Lobato, and R. Turner. Black-box alpha divergence minimization. In *International conference on machine learning*, pages 1511–1520. PMLR, 2016.

[140] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

[141] G. E. Hinton and R. Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in Neural Information Processing Systems*, 6, 1993.

[142] H. Hoang, P. De Mey, and O. Talagrand. 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, pages 747–752 vol.1, Dec. 1994. doi: 10.1109/CDC.1994.410863.

[143] S. Hoang, R. Baraille, O. Talagrand, X. Carton, and P. De Mey. Adaptive filtering: Application to satellite data assimilation in oceanography. *Dynamics of Atmospheres and Oceans*, 27(1):257–281, Jan. 1998. ISSN 0377-0265. doi: 10.1016/S0377-0265(97)00014-6.

[144] T.-V. Hoang, S. Krumscheid, H. G. Matthies, and R. Tempone. Machine learning-based conditional mean filter: a generalization of the ensemble Kalman filter for nonlinear data assimilation. *arXiv preprint arXiv:2106.07908*, 2021.

[145] H. Hoel, K. J. H. Law, and R. Tempone. Multilevel ensemble Kalman filtering. *SIAM Journal on Numerical Analysis*, 54(3):1813–1839, Jan. 2016. ISSN 0036-1429. doi: 10.1137/15M100955X.

[146] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

[147] F. Hoffmann, B. Hosseini, A. A. Oberai, and A. M. Stuart. Spectral analysis of weighted Laplacians arising in data clustering. *Applied and Computational Harmonic Analysis*, 56:189–249, 2022.

[148] M. Holland and I. Melbourne. Central limit theorems and invariance principles for Lorenz attractors. *Journal of the London Mathematical Society*, 76(2):345–364, 2007.

[149] B. Hosseini. Well-posed Bayesian inverse problems with infinitely divisible and heavy-tailed prior measures. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):1024–1060, 2017.

[150] B. Hosseini and N. Nigam. Well-posed Bayesian inverse problems: Priors with exponential tails. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):436–465, 2017.

[151] B. Hosseini, A. W. Hsu, and A. Taghvaei. Conditional optimal transport on function spaces. *arXiv preprint arXiv:2311.05672*, 2023.

[152] D. Z. Huang and J. Huang. Unscented Kalman inversion: efficient Gaussian approximation to the posterior distribution. *arXiv preprint arXiv:2103.00277*, 2021.

[153] D. Z. Huang, T. Schneider, and A. M. Stuart. Unscented Kalman inversion. *arXiv preprint arXiv:2102.01580*, 2021.

[154] D. Z. Huang, T. Schneider, and A. M. Stuart. Iterated Kalman methodology for inverse problems. *Journal of Computational Physics*, 463:111262, 2022.

[155] B. R. Hunt, E. J. Kostelich, and I. Szunyogh. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena*, 230(1):112–126, June 2007. ISSN 0167-2789. doi: 10.1016/j.physd.2006.11.008.

[156] A. Hyvärinen and P. Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[157] M. A. Iglesias. Iterative regularization for ensemble data assimilation in reservoir models. *Computational Geosciences*, 19(1):177–212, 2015.

[158] M. A. Iglesias. A regularizing iterative ensemble Kalman method for PDE-constrained inverse problems. *Inverse Problems*, 32(2):025002, 2016.

[159] M. A. Iglesias and Y. Yang. Adaptive regularisation for ensemble Kalman inversion. *Inverse Problems*, 37(2):025008, 2021.

[160] M. A. Iglesias, K. J. H. Law, and A. M. Stuart. Ensemble Kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, 2014.

[161] T. Ishizone, T. Higuchi, and K. Nakamura. Ensemble Kalman variational objectives: nonlinear latent trajectory inference with a hybrid of variational inference and ensemble Kalman filter. *arXiv preprint arXiv:2010.08729*, 2020.

[162] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[163] A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* Number 64 in Mathematics in Science and Engineering. Academic Press, Inc., New York, 1970.

[164] A. Jordan, F. Krüger, and S. Lerch. Evaluating probabilistic forecasts with scoring rules. *Journal of Statistical Software*, 90:1–37, Aug. 2019. ISSN 1548-7660. doi: 10.18637/jss.v090.i12.

[165] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

[166] J. Kaipio and E. Somersalo. Statistical and Computational Inverse Problems. *Springer Science & Business Media*, 160, 2006.

[167] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[168] E. Kalnay. *Atmospheric Modeling, Data Assimilation and Predictability.* Cambridge University Press, New York, Dec. 2002. ISBN 978-0-521-79629-3.

[169] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, Y. Zhu, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K. C. Mo, C. Ropelewski, J. Wang, A. Leetmaa, R. Reynolds, R. Jenne, and D. Joseph. The NCEP/NCAR 40-Year Reanalysis Project. *Bulletin of the American Meteorological Society*, 77(3):437–472, Mar. 1996. ISSN 0003-0007, 1520-0477. doi: 10.1175/1520-0477(1996)077<0437:TNYRP>2.0.CO;2.

[170] E. Kalnay, T. Sluka, T. Yoshida, C. Da, and S. Mote. Review article: Towards strongly coupled ensemble data assimilation with additional improvements from machine learning. *Nonlinear Processes in Geophysics*, 30(2):217–236, June 2023. ISSN 1023-5809. doi: 10.5194/npg-30-217-2023.

[171] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[172] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in Neural Information Processing Systems*, 31, 2018.

[173] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations*, 2014.

[174] D. P. Kingma, M. Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[175] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.

[176] R. Krishnan, U. Shalit, and D. Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[177] F. Laio and S. Tamea. Verification tools for probabilistic forecasts of continuous hydrological variables. *Hydrology and Earth System Sciences*, 11(4):1267–1277, 2007.

[178] M. Lambert, S. Chewi, F. Bach, S. Bonnabel, and P. Rigollet. Variational inference via wasserstein gradient flows. *Advances in Neural Information Processing Systems*, 35:14434–14447, 2022.

[179] M. Lambert, S. Bonnabel, and F. Bach. Variational Gaussian approximation of the Kushner optimal filter. In F. Nielsen and F. Barbaresco, editors, *Geometric Science of Information*, Lecture Notes in Computer Science, pages 395–404, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-38271-0. doi: 10.1007/978-3-031-38271-0_39.

[180] H. Lambley and T. Sullivan. An order-theoretic perspective on modes and maximum a posteriori estimation in Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 11(4):1195–1224, 2023.

[181] P. Lancaster and L. Rodman. *Algebraic Riccati Equations.* Clarendon press, 1995.

[182] K. Lange. *MM Optimization Algorithms.* SIAM, 2016.

[183] S. Lasanen. Non-Gaussian statistical inverse problems. Part I: Posterior distributions. *Inverse Problems & Imaging*, 6(2):215–266, 2012.

[184] S. Lasanen. Non-Gaussian statistical inverse problems. Part II: Posterior convergence for approximated unknowns. *Inverse Problems & Imaging*, 6(2):267, 2012.

[185] J. Latz. On the well-posedness of Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 8(1):451–482, 2020.

[186] R. S. Laugesen, P. G. Mehta, S. P. Meyn, and M. Raginsky. Poisson's equation in nonlinear filtering. *SIAM Journal on Control and Optimization*, 53(1):501–525, Jan. 2015. ISSN 0363-0129. doi: 10.1137/13094743X.

[187] K. J. H. Law, A. Shukla, and A. M. Stuart. Analysis of the 3DVAR filter for the partially observed Lorenz'63 model. *Discrete and Continuous Dynamical Systems*, 34(3):1061–1078, 2014.

[188] K. J. H. Law, A. M. Stuart, and K. Zygalakis. *Data Assimilation.* Springer, 2015.

[189] T. A. Le, M. Igl, T. Rainforth, T. Jin, and F. Wood. Auto-encoding sequential Monte Carlo. *arXiv preprint arXiv:1705.10306*, 2017.

[190] F. Le Gland, V. Monbet, and V. Tran. Large sample asymptotics for the ensemble Kalman filter. *PhD Thesis*, 2009.

[191] H. Lee, J. Lu, and Y. Tan. Convergence of score-based generative modeling for general data distributions. In *International Conference on Algorithmic Learning Theory*, pages 946–985. PMLR, 2023.

[192] J. Lei and P. Bickel. A moment matching ensemble filter for nonlinear non-Gaussian data assimilation. *Monthly Weather Review*, 139(12):3964–3973, 2011.

[193] M. Leutbecher and T. Haiden. Understanding changes of the continuous ranked probability score using a homogeneous Gaussian approximation. *Quarterly Journal of the Royal Meteorological Society*, 147(734):425–442, 2021. ISSN 1477-870X. doi: 10.1002/qj.3926.

[194] M. Levine and A. Stuart. 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.

[195] R. Lguensat, P. Tandeo, P. Ailliot, M. Pulido, and R. Fablet. The Analog Data Assimilation. *Monthly Weather Review*, 145(10):4093–4107, Aug. 2017. ISSN 0027-0644. doi: 10.1175/MWR-D-16-0441.1.

[196] L. Li, R. Carver, I. Lopez-Gomez, F. Sha, and J. Anderson. SEEDS: Emulation of weather forecast ensembles with diffusion models. *arXiv preprint arXiv:2306.14066*, 2023.

[197] Y. Li and R. E. Turner. Rényi divergence variational inference. *Advances in Neural Information Processing Systems*, 29, 2016.

[198] F. Liang, M. Mahoney, and L. Hodgkinson. Fat–tailed variational inference with anisotropic tail adaptive flows. In *International Conference on Machine Learning*, pages 13257–13270. PMLR, 2022.

[199] Q. Liu and D. Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in Neural Information Processing Systems*, 29, 2016.

[200] I. Lopez-Gomez, C. Christopoulos, H. L. Langeland Ervik, O. R. Dunbar, Y. Cohen, and T. Schneider. Training physics-based machine-learning parameterizations with gradient-free ensemble Kalman methods. *Journal of Advances in Modeling Earth Systems*, 14(8):e2022MS003105, 2022.

[201] A. Lorenc. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474):1177–1194, 1986.

[202] E. N. Lorenz. Atmospheric predictability as revealed by naturally occurring analogues. *Journal of the Atmospheric Sciences*, 26(4):636–646, July 1969. ISSN 0022-4928. doi: 10.1175/1520-0469(1969)26<636:APARBN>2.0.CO;2.

[203] E. Luk, E. Bach, R. Baptista, and A. Stuart. Learning optimal filters using variational inference. *arXiv preprint arXiv:2406.18066*, 2024.

[204] S. Lunz, O. Öktem, and C.-B. Schönlieb. Adversarial regularizers in inverse problems. *Advances in Neural Information Processing Systems*, 31, 2018.

[205] C. J. Maddison, D. Lawson, G. Tucker, N. Heess, M. Norouzi, A. Mnih, A. Doucet, and Y. W. Teh. Filtering variational objectives. *arXiv preprint arXiv:1705.09279*, 2017.

[206] A. J. Majda and J. Harlim. *Filtering Complex Turbulent Systems*. Cambridge University Press, 2012.

[207] N. Mallia-Parfitt and J. Bröcker. Assessing the performance of data assimilation algorithms which employ linear error feedback. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(10):103109, Oct. 2016. ISSN 1054-1500. doi: 10.1063/1.4965029.

[208] J. Mandel, L. Cobb, and J. D. Beezley. On the convergence of the ensemble Kalman filter. *Applications of Mathematics*, 56(6):533–541, 2011.

[209] J. Marino, M. Cvitkovic, and Y. Yue. A General Method for Amortizing Variational Filtering. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[210] J. Marino, Y. Yue, and S. Mandt. Iterative amortized inference. In *International Conference on Machine Learning*, pages 3403–3412. PMLR, 2018.

[211] Y. Marzouk and D. Xiu. A stochastic collocation approach to Bayesian inference in inverse problems. *Communications in Computational Physics*, 6(4):826–847, 2009.

[212] Y. Marzouk, T. Moselhy, M. Parno, and A. Spantini. Sampling via measure transport: An introduction. *Handbook of Uncertainty Quantification*, pages 1–41, 2016.

[213] J. E. Matheson and R. L. Winkler. Scoring rules for continuous probability distributions. *Management Science*, 22(10):1087–1096, 1976. ISSN 0025-1909.

[214] A. Maurais, T. Alsup, B. Peherstorfer, and Y. Marzouk. Multi-Fidelity Covariance Estimation in the Log-Euclidean Geometry. June 2023.

[215] A. Maurais, T. Alsup, B. Peherstorfer, and Y. Marzouk. Multifidelity Covariance Estimation via Regression on the Manifold of Symmetric Positive Definite Matrices, July 2023.

[216] P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 3. Academic Press, New York, 1982. ISBN 0-12-480703-8.

[217] M. McCabe and J. Brown. Learning to assimilate in chaotic dynamical systems. In *Advances in Neural Information Processing Systems*, volume 34, pages 12237–12250. Curran Associates, Inc., 2021.

[218] X.-L. Meng and D. Van Dyk. The EM algorithm—an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 59(3):511–567, 1997.

[219] S. Metref, A. Hannart, J. Ruiz, M. Bocquet, A. Carrassi, and M. Ghil. Estimating model evidence using ensemble-based data assimilation with localization–The model selection problem. *Quarterly Journal of the Royal Meteorological Society*, 145(721):1571–1588, 2019.

[220] E. L. Miller and W. C. Karl. Fundamentals of Inverse Problems. *Not yet published*, 2003.

[221] T. Miyoshi. 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.

[222] A. J. Moodey, A. S. Lawless, R. W. Potthast, and P. J. Van Leeuwen. Nonlinear error dynamics for cycled data assimilation methods. *Inverse Problems*, 29(2): 025002, 2013.

[223] A. Moosavi, A. Attia, and A. Sandu. Tuning Covariance Localization Using Machine Learning. In J. M. F. Rodrigues, P. J. S. Cardoso, J. Monteiro, R. Lam, V. V. Krzhizhanovskaya, M. H. Lees, J. J. Dongarra, and P. M. Sloot, editors, *Computational Science – ICCS 2019*, Lecture Notes in Computer Science, pages 199–212, Cham, 2019. Springer International Publishing. ISBN 978-3-030-22747-0. doi: 10.1007/978-3-030-22747-0_16.

[224] K. Muandet, K. Fukumizu, B. Sriperumbudur, B. Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

[225] M. Mureşan. *A Concrete Approach to Classical Analysis.* CMS Books in Mathematics. Springer, New York, NY, 2009. ISBN 978-0-387-78932-3 978-0-387-78933-0. doi: 10.1007/978-0-387-78933-0.

[226] A. H. Murphy and E. S. Epstein. Skill scores and correlation coefficients in model verification. *Monthly weather review*, 117(3):572–582, 1989.

[227] C. Naesseth, S. Linderman, R. Ranganath, and D. Blei. Variational sequential Monte Carlo. In *International Conference on Artificial Intelligence and Statistics*, pages 968–977. PMLR, 2018.

[228] A. Narayan, Y. Marzouk, and D. Xiu. Sequential data assimilation with multiple models. *Journal of Computational Physics*, 231(19):6401–6418, Aug. 2012. ISSN 0021-9991. doi: 10.1016/j.jcp.2012.06.002.

[229] D. G. Nel. On the symmetric multivariate normal distribution and the asymptotic expansion of a Wishart matrix. *South African Statistical Journal*, 12(2):145–159, Jan. 1978. doi: 10.10520/AJA0038271X_302.

[230] N. H. Nelsen and A. M. Stuart. The random feature model for input-output maps between Banach spaces. *SIAM Journal on Scientific Computing*, 43(5): A3212–A3243, 2021.

[231] Y. Nesterov. *Introductory Lectures on Convex Optimization*, volume 87 of *Applied Optimization*. Springer US, Boston, MA, 2004. ISBN 978-1-4613-4691-3 978-1-4419-8853-9. doi: 10.1007/978-1-4419-8853-9.

[232] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14, 2001.

[233] D. Nguyen, S. Ouala, L. Drumetz, and R. Fablet. Em-like learning chaotic dynamics from noisy and partial observations. *arXiv preprint arXiv:1903.10335*, 2019.

[234] R. Nickl. *Bayesian Non-linear Statistical Inverse Problems*. 2022. URL http://www.statslab.cam.ac.uk/~nickl/Site/__files/lecturenotes.pdf.

[235] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, New York, 2006. ISBN 978-0-387-30303-1. doi: 10.1007/978-0-387-40065-5.

[236] M. Oczkowski, I. Szunyogh, and D. J. Patil. Mechanisms for the development of locally low-dimensional atmospheric dynamics. *Journal of the Atmospheric Sciences*, 62(4):1135–1156, Apr. 2005. ISSN 0022-4928, 1520-0469. doi: 10.1175/JAS3403.1.

[237] H. R. Ogrosky, S. N. Stechmann, N. Chen, and A. J. Majda. Singular Spectrum Analysis With Conditional Predictions for Real-Time State Estimation and Forecasting. *Geophysical Research Letters*, 46(3):1851–1860, Feb. 2019. ISSN 0094-8276, 1944-8007. doi: 10.1029/2018GL081100.

[238] J. Oliva, A. Dubey, M. Zaheer, B. Poczos, R. Salakhutdinov, E. Xing, and J. Schneider. Transformation autoregressive networks. In *International Conference on Machine Learning*, pages 3898–3907. PMLR, 2018.

[239] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

[240] M. Opper and C. Archambeau. The variational Gaussian approximation revisited. *Neural Computation*, 21(3):786–792, 2009.

[241] H. Owhadi and C. Scovel. *Operator-Adapted Wavelets, Fast Solvers, and Numerical Homogenization: From a Game Theoretic Approach to Numerical Approximation and Algorithm Design*, volume 35. Cambridge University Press, 2019.

[242] H. A. Panofsky. OBJECTIVE WEATHER-MAP ANALYSIS. *Journal of Meteorology*, 6(6):386–392, Dec. 1949. ISSN 1520-0469. doi: 10.1175/1520-0469(1949)006<0386:OWMA>2.0.CO;2.

[243] G. Papamakarios, D. Sterratt, and I. Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 837–848. PMLR, 2019.

[244] D. V. Patel and A. A. Oberai. GAN-Based Priors for Quantifying Uncertainty in Supervised Learning. *SIAM/ASA Journal on Uncertainty Quantification*, 9(3):1314–1343, 2021.

[245] D. J. Patil, B. R. Hunt, E. Kalnay, J. A. Yorke, and E. Ott. Local low dimensionality of atmospheric dynamics. *Physical Review Letters*, 86(26):5878–5881, June 2001. doi: 10.1103/PhysRevLett.86.5878.

[246] B. Peherstorfer, K. Willcox, and M. Gunzburger. Optimal model management for multifidelity Monte Carlo estimation. *SIAM Journal on Scientific Computing*, 38(5):A3163–A3194, Jan. 2016. ISSN 1064-8275. doi: 10.1137/15M1046472.

[247] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization. *SIAM Review*, 60(3):550–591, Jan. 2018. ISSN 0036-1445. doi: 10.1137/16M1082469.

[248] S. G. Penny, T. A. Smith, T.-C. Chen, J. A. Platt, H.-Y. Lin, M. Goodliff, and H. D. I. Abarbanel. Integrating recurrent neural networks with data assimilation for scalable data-driven state estimation. *Journal of Advances in Modeling Earth Systems*, 14(3):e2021MS002843, 2022. ISSN 1942-2466. doi: 10.1029/2021MS002843.

[249] G. Peyré, M. Cuturi, et al. Computational optimal transport: with applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[250] A. Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8:143–195, 1999.

[251] J. A. Platt, S. G. Penny, T. A. Smith, T.-C. Chen, and H. D. I. Abarbanel. Constraining chaos: Enforcing dynamical invariants in the training of reservoir computers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(10):103107, Oct. 2023. ISSN 1054-1500. doi: 10.1063/5.0156999.

[252] A. A. Popov and A. Sandu. A Bayesian approach to multivariate adaptive localization in ensemble-based data assimilation with time-dependent extensions. *Nonlinear Processes in Geophysics*, 26(2):109–122, June 2019. ISSN 1023-5809. doi: 10.5194/npg-26-109-2019.

[253] A. A. Popov, C. Mou, A. Sandu, and T. Iliescu. A multifidelity ensemble Kalman filter with reduced order control variates. *SIAM Journal on Scientific Computing*, 43(2):A1134–A1162, Jan. 2021. ISSN 1064-8275. doi: 10.1137/20M1349965.

[254] I. Price, A. Sanchez-Gonzalez, F. Alet, T. Ewalds, A. El-Kadi, J. Stott, S. Mohamed, P. Battaglia, R. Lam, and M. Willson. GenCast: Diffusion-based ensemble forecasting for medium-range weather. *arXiv preprint arXiv:2312.15796*, 2023.

[255] M. L. Provost, R. Baptista, J. D. Eldredge, and Y. Marzouk. An adaptive ensemble filter for heavy-tailed distributions: tuning-free inflation and localization. *arXiv preprint arXiv:2310.08741*, 2023.

[256] M. Pulido and P. J. van Leeuwen. Sequential monte carlo with kernel embedded mappings: The mapping particle filter. *Journal of Computational Physics*, 396: 400–415, 2019.

[257] M. Pulido, P. Tandeo, M. Bocquet, A. Carrassi, and M. Lucini. Stochastic parameterization identification using ensemble Kalman filtering combined with maximum likelihood methods. *Tellus A: Dynamic Meteorology and Oceanography*, 70(1):1–17, 2018.

[258] A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 2007.

[259] A. Rahimi and B. Recht. Uniform approximation of functions with random bases. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 555–561. IEEE, 2008.

[260] A. Rahimi and B. Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in Neural Information Processing Systems*, 21, 2008.

[261] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. *arXiv preprint arXiv:1801.01236*, 2018.

[262] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378: 686–707, 2019.

[263] R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

[264] S. S. Rangapuram, M. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski. Deep state space models for time series forecasting. In *Proceedings of the 32nd international conference on neural information processing systems*, pages 7796–7805, 2018.

[265] S. Rasp, M. S. Pritchard, and P. Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, 2018.

[266] K. Rasul, A.-S. Sheikh, I. Schuster, U. Bergmann, and R. Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. *International Conference on Learning Representations*, 2021.

[267] D. Ray, J. Murgoitio-Esandi, A. Dasgupta, and A. A. Oberai. Solution of physics-based inverse problems using conditional generative adversarial networks with full gradient penalty. *arXiv preprint arXiv:2306.04895*, 2023.

[268] S. Reich. Data assimilation: the Schrödinger perspective. *Acta Numerica*, 28: 635–711, 2019.

[269] S. Reich and C. Cotter. *Probabilistic Forecasting and Bayesian Data Assimilation.* Cambridge University Press, 2015.

[270] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[271] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[272] W. Sacher and P. Bartello. Sampling errors in ensemble Kalman filtering. Part I: Theory. *Monthly Weather Review*, 136(8):3035–3049, Aug. 2008. ISSN 0027-0644. doi: 10.1175/2007MWR2323.1.

[273] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.

[274] S. K. Sahu and G. O. Roberts. On convergence of the EM algorithmand the Gibbs sampler. *Statistics and Computing*, 9(1):55–64, 1999.

[275] D. Sanz-Alonso. Importance sampling and necessary sample size: An information theory approach. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):867–879, 2018.

[276] D. Sanz-Alonso and Z. Wang. Bayesian update with importance sampling: Required sample size. *Entropy*, 23(1):22, 2021.

[277] D. Sanz-Alonso and R. Yang. The SPDE approach to Matérn fields: graph representations. *arXiv preprint arXiv:2004.08000*, 2020.

[278] D. Sanz-Alonso, A. Stuart, and A. Taeb. *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.

[279] K. Sargsyan, X. Huan, and H. N. Najm. Embedded model error representation for Bayesian model calibration. *International Journal for Uncertainty Quantification*, 9(4), 2019.

[280] S. Särkkä. *Bayesian Filtering and Smoothing*, volume 3. Cambridge University Press, 2013.

[281] T. Sauer, J. A. Yorke, and M. Casdagli. Embedology. *Journal of Statistical Physics*, 65(3):579–616, Nov. 1991. ISSN 1572-9613. doi: 10.1007/BF01053745.

[282] D. Schaden and E. Ullmann. On Multilevel Best Linear Unbiased Estimators. *SIAM/ASA Journal on Uncertainty Quantification*, 8(2):601–635, Jan. 2020. doi: 10.1137/19M1263534.

[283] S. Scher and G. Messori. Ensemble Methods for Neural Network-Based Weather Forecasts. *Journal of Advances in Modeling Earth Systems*, 13(2), 2021. ISSN 1942-2466. doi: 10.1029/2020MS002331.

[284] C. Schillings and A. M. Stuart. Analysis of the ensemble Kalman filter for inverse problems. *SIAM Journal on Numerical Analysis*, 55(3):1264–1290, 2017.

[285] C. Schillings and A. M. Stuart. Convergence analysis of ensemble kalman inversion: the linear, noisy case. *Applicable Analysis*, 97(1):107–123, 2018.

[286] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[287] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The annals of statistics*, pages 2263–2291, 2013.

[288] B. W. Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[289] S. A. Sisson, Y. Fan, and M. Beaumont. *Handbook of approximate Bayesian computation*. CRC Press, 2018.

[290] A. J. Smola and B. Schölkopf. *Learning With Kernels*, volume 4. Citeseer, 1998.

[291] C. Snyder, T. Bengtsson, P. Bickel, and J. L. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2016.

[292] Y. S. Soh and V. Chandrasekaran. Learning semidefinite regularizers. *Foundations of Computational Mathematics*, 19(2):375–434, 2019.

[293] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[294] A. Spantini, R. Baptista, and Y. Marzouk. Coupling techniques for nonlinear ensemble filtering. *SIAM Review*, 64(4):921–953, 2022.

[295] B. Sprungk. On the local lipschitz stability of bayesian inverse problems. *Inverse Problems*, 36(5):055015, 2020.

[296] B. K. Sriperumbudur, K. Fukumizu, and G. R. Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 12(7), 2011.

[297] J. R. Stroud and T. Bengtsson. Sequential state and variance estimation within the ensemble Kalman filter. *Monthly Weather Review*, 135(9):3194–3208, 2007.

[298] J. R. Stroud, M. L. Stein, B. M. Lesht, D. J. Schwab, and D. Beletsky. An ensemble Kalman filter and smoother for satellite data assimilation. *Journal of the American Statistical Association*, 105(491):978–990, 2010.

[299] J. R. Stroud, M. Katzfuss, and C. K. Wikle. A Bayesian adaptive ensemble Kalman filter for sequential state and parameter estimation. *Monthly Weather Review*, 146(1):373–386, 2018.

[300] A. Stuart and A. Teckentrup. Posterior consistency for Gaussian process approximations of Bayesian posterior distributions. *Mathematics of Computation*, 87 (310):721–753, 2018.

[301] A. M. Stuart. Inverse problems: a Bayesian perspective. *Acta Numerica*, 19: 451–559, 2010.

[302] H. Sun and K. L. Bouman. Deep probabilistic imaging: Uncertainty quantification and multi-modal solution characterization for computational imaging. *arXiv preprint arXiv:2010.14462*, 9, 2020.

[303] T. Sutter, A. Ganguly, and H. Koeppl. A variational approach to path estimation and parameter inference of hidden diffusion processes. *Journal of Machine Learning Research*, 17(190):1–37, 2016. ISSN 1533-7928.

[304] G. J. Székely and M. L. Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, Aug. 2013. ISSN 0378-3758. doi: 10.1016/j.jspi.2013.03.018.

[305] E. G. Tabak and E. Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

[306] A. Taghvaei and B. Hosseini. An optimal transport formulation of Bayes' law for nonlinear filtering algorithms. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 6608–6613. IEEE, 2022.

[307] P. Tandeo, M. Pulido, and F. Lott. Offline parameter estimation using EnKF and maximum likelihood error covariance estimates: Application to a subgrid-scale orography parametrization. *Quarterly journal of the royal meteorological society*, 141(687):383–395, 2015.

[308] P. Tandeo, P. Ailliot, M. Bocquet, A. Carrassi, T. Miyoshi, M. Pulido, and Y. Zhen. A Review of Innovation-Based Methods to Jointly Estimate Model and Observation Error Covariance Matrices in Ensemble Data Assimilation. *Monthly Weather Review*, 148(10):3973–3994, Sept. 2020. ISSN 1520-0493, 0027-0644. doi: 10.1175/MWR-D-19-0240.1.

[309] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation.* SIAM, 2015.

[310] J. Thorey, V. Mallet, and P. Baudin. Online learning with the continuous ranked probability score for ensemble forecasting. *Quarterly Journal of the Royal Meteorological Society*, 143(702):521–529, 2017. ISSN 1477-870X. doi: 10.1002/qj.2940.

[311] A. N. Tikhonov and V. Y. Arsenin. Solutions of Ill-posed Problems. *Washington, Winston & Sons*, 1977.

[312] M. K. Tippett, J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker. Ensemble square root filters. *Monthly Weather Review*, 131(7):1485–1490, 2003.

[313] W. Tucker. The Lorenz attractor exists. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 328(12):1197–1202, 1999.

[314] W. Tucker. A rigorous ODE solver and Smale's 14th problem. *Foundations of Computational Mathematics*, 2:53–117, 2002.

[315] G. Ueno and N. Nakamura. Iterative algorithm for maximum-likelihood estimation of the observation-error covariance matrix for ensemble-based filters. *Quarterly Journal of the Royal Meteorological Society*, 140(678):295–315, 2014.

[316] G. Ueno and N. Nakamura. Bayesian estimation of the observation-error covariance matrix in ensemble-based filters. *Quarterly Journal of the Royal Meteorological Society*, 142(698):2055–2080, 2016.

[317] S. M. Uppala, P. Kållberg, A. J. Simmons, U. Andrae, V. D. C. Bechtold, M. Fiorino, J. Gibson, J. Haseler, A. Hernandez, G. Kelly, et al. The ERA-40 re-analysis. *Quarterly Journal of the Royal Meteorological Society: A Journal of the Atmospheric Sciences, Applied Meteorology and Physical Oceanography*, 131 (612):2961–3012, 2005.

[318] A. Vadeboncoeur, Ö. D. Akyildiz, I. Kazlauskaite, M. Girolami, and F. Cirak. Fully probabilistic deep models for forward and inverse problems in parametric PDEs. *Journal of Computational Physics*, 491:112369, 2023.

[319] R. Vautard, P. Yiou, and M. Ghil. Singular-spectrum analysis: A toolkit for short, noisy chaotic signals. *Physica D: Nonlinear Phenomena*, 58(1):95–126, Sept. 1992. ISSN 0167-2789. doi: 10.1016/0167-2789(92)90103-T.

[320] C. Villani. *Optimal Transport: Old and New*, volume 338. Springer, 2009.

[321] P. Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

[322] D. Vishny, M. Morzfeld, K. Gwirtz, E. Bach, O. R. A. Dunbar, and D. Hodyss. High-Dimensional Covariance Estimation From a Small Number of Samples. *Journal of Advances in Modeling Earth Systems*, 16(9):e2024MS004417, Aug. 2024. ISSN 1942-2466. doi: 10.1029/2024MS004417.

[323] C. R. Vogel. *Computational Methods for Inverse Problems.* SIAM, 2002.

[324] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17 (4):395–416, 2007.

[325] U. Von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008.

[326] M. D. Vrettas, D. Cornford, and M. Opper. Estimating parameters in stochastic systems: A variational Bayesian approach. *Physica D: Nonlinear Phenomena*, 240 (23):1877–1900, 2011.

[327] M. D. Vrettas, M. Opper, and D. Cornford. Variational mean-field algorithm for efficient inference in large systems of stochastic differential equations. *Physical Review E*, 91(1):012148, 2015.

[328] G. Wahba. *Spline Models for Observational Data.* SIAM, 1990.

[329] M. J. Wainwright, M. I. Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2): 1–305, 2008.

[330] Z. Wang, L. Lei, J. L. Anderson, Z.-M. Tan, and Y. Zhang. Convolutional Neural Network-Based Adaptive Localization for an Ensemble Kalman Filter. *Journal of Advances in Modeling Earth Systems*, 15(10):e2023MS003642, 2023. ISSN 1942-2466. doi: 10.1029/2023MS003642.

[331] G. C. Wei and M. A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704, 1990.

[332] H. Wendland. *Scattered Data Approximation*, volume 17. Cambridge University Press, 2004.

[333] D. S. Wilks. *Statistical Methods in the Atmospheric Sciences*. Elsevier, 4 edition, 2019. ISBN 978-0-12-815823-4. doi: 10.1016/C2017-0-03921-6.

[334] C. K. Williams and C. E. Rasmussen. *Gaussian Processes for Machine Learning*, volume 2. MIT press Cambridge, MA, 2006.

[335] C. Winkler, D. Worrall, E. Hoogeboom, and M. Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

[336] J.-L. Wu, M. E. Levine, T. Schneider, and A. Stuart. Learning about structural errors in models of complex dynamical systems. *arXiv preprint arXiv:2401.00035*, 2023.

[337] Y. Xiao, L. Bai, W. Xue, K. Chen, T. Han, and W. Ouyang. FengWu-4DVar: Coupling the Data-driven Weather Forecasting Model with 4D Variational Assimilation. https://arxiv.org/abs/2312.12455v1, Dec. 2023.

[338] A. Zellner. Optimal information processing and Bayes's theorem. *The American Statistician*, 42(4):278–280, 1988.

[339] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt. Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41 (8):2008–2026, 2018.

# Alphabetical Index