Doctoral theses at NTNU, 2021:303

Margrethe Kvale Loe

Ensemble updating for a state-space model with categorical variables

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Information Technology and Electrical
Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

Margrethe Kvale Loe

Ensemble updating for a state-space model with categorical variables

Thesis for the Degree of Philosophiae Doctor

Trondheim, September 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

ISBN 978-82-326-6950-9 (printed ver.)
ISBN 978-82-326-5252-5 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2021:303

Printed by NTNU Grafisk senter

Acknowledgements

First of all, I would like to thank my supervisor, Professor Håkon Tjelmeland, for his excellent guidance throughout my work on this thesis. I could not have asked for a greater supervisor. I would also like to thank my number one support system: my parents. Without their love and support, I would never have completed this work. Thanks to the administrative staff, especially Marte Pernille Hatlo Andresen and (former employee) Anne Kajander, for the understanding and kindness I was shown during challenging times. Thanks to the professors and Ph.D. students of the Uncertainty in Reservoir Evaluation research initiative, for many interesting discussions, some of which related to research. Finally, thanks to my two office mates the last couple of years, Ola Mæhlen and Susan Anyosa, for fun conversations, cute drawings on the whiteboard, and for making room 1056 a good place to study and a great place to be.

Margrethe Kvale Loe
Trondheim
May, 2021

Thesis outline

Background

Paper 1: **Ensemble updating of binary state vectors by maximising the expected number of unchanged components**
Margrethe Kvale Loe and Håkon Tjelmeland
To appear in *Scandinavian Journal of Statistics*

Paper 2: **Geophysics-based fluid-facies predictions using ensemble updating of binary state vectors**
Margrethe Kvale Loe, Dario Grana and Håkon Tjelmeland
Published in *Mathematical Geosciences*

Paper 3: **A generalised and fully Bayesian ensemble updating framework**
Margrethe Kvale Loe and Håkon Tjelmeland
Technical report

Paper 4: **Ensemble updating of categorical state vectors**
Margrethe Kvale Loe and Håkon Tjelmeland
Submitted

Background

1 Introduction

A problem which arises in many real-life scientific applications is the need to make inference about an unobserved dynamic process based on a series of indirect, noisy measurements. Examples include applications in control engineering, finance and economics, and several fields of the geosciences. Often in such situations, a so-called state-space model is adopted. The unobserved dynamic process is then modelled as a sequence of states, or state vectors, that evolve in time according to a first-order Markov chain, and the observations are assumed to be conditionally independent given the states. The problem of making inference about the unobserved state at a certain point in time, given all observations available at this time, is known as filtering.

Filtering is a recursive inference procedure which allows observations to be assimilated sequentially as they arrive. From a Bayesian perspective, each iteration involves a standard Bayesian inference problem, where the goal is to compute a posterior model, in this context called the filtering distribution, by conditioning a prior model, in this context called the forecast distribution, on new observations assumed to be distributed according to a corresponding likelihood model. In the special case of a linear-Gaussian state-space model, the recursion leads to the famous Kalman filter (Kalman, 1960). In most situations, however, exact computation of the filtering distributions is problematic due to complex and/or high-dimensional integrals. Approximate solutions are therefore required, and approaches based on simulation, so-called *ensemble methods*, where the filtering distributions instead are empirically represented with an ensemble of realisations, are usually the best option.

The focus of the present thesis is filtering of high-dimensional state vectors where each element is a *categorical* variable. The integrals in each step of the filtering recursions then represent summations which, brute force, are too computationally demanding to cope with. For state-space models with continuous state and observation vectors, there is an extensive literature on ensemble-based filtering methods, of which the most important contribution perhaps is the ensemble Kalman filter as presented in Burgers et al. (1998). Filtering of categorical state vectors, however, has received considerably less attention. Particle filters (Gordon et al., 1993; Doucet et al., 2001) are in principle applicable, but break down in high-dimensional situations. The main objective of the present thesis is to develop a novel and computationally feasible ensemble filtering method for high-dimensional, categorical state vectors.

2 State-space models

To motivate the work of this thesis, we now review state-space models and the associated filtering problem in more detail. State-space models are also known as hidden Markov models (HMMs), although this term is often reserved for models with categorical states (Künsch, 2000).

2.1 The general state-space model

A state-space model is a probabilistic model consisting of an unobserved discrete-time stochastic process $\{x_t, t \in \mathbb{N}\}$ and a corresponding observed discrete-time stochastic process $\{y_t, t \in \mathbb{N}\}$ where y_t is a partial observation of x_t at time t. It is possible to extend the following material to situations where an observation y_t is only available at a few of the time steps $t = 1, 2, \dots$, but for simplicity we restrict here the focus to situations where an observation y_t is available at every time step. Each x_t is an n-dimensional vector taking value in a sample space $\Omega_x \subseteq \mathbb{R}^n$ and each y_t is an m-dimensional vector taking value in a sample space $\Omega_y \subseteq \mathbb{R}^m$. The unobserved x_t-process is called the state process, and the vector x_t is called the state vector, or simply the state, at time t. For notational simplicity, we will in the following use the notations $x_{s:t} = (x_s, \dots, x_t)$ and $y_{s:t} = (y_s, \dots, y_t)$ to denote the vector of states and the vector of observations, respectively, from time s to time t, for $s \leq t$. In the state-space representation, the unobserved state process is assumed to evolve in time according to a first-order Markov chain with initial distribution $p(x_1)$ and transition probabilities $p(x_t|x_{t-1})$, i.e.

$$p(x_{1:t}) = p(x_1) \prod_{j=2}^{t} p(x_j|x_{j-1}). \tag{1}$$

The observations $y_1, y_2, \dots$ of the observed process are assumed to be conditionally independent given $\{x_t, t \in \mathbb{N}\}$, with y_t depending only on x_t. This means that the joint likelihood of $y_{1:t}$ given $x_{1:t}$ is given as

$$p(y_{1:t}|x_{1:t}) = \prod_{j=1}^{t} p(y_j|x_j). \tag{2}$$

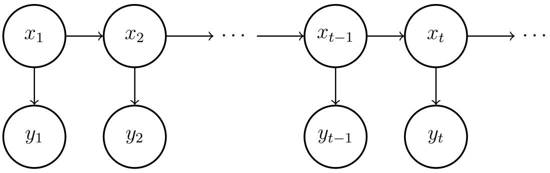


Figure 1: Graphical illustration of a state-space model

For continuous state and observation vectors, an equivalent, and very common, way to specify a state-space model is as a discrete-time dynamical system, with initial conditions $x_1 \sim p(x_1)$, a state equation,

$$x_t = f(t, x_{t-1}, \varepsilon_t), \tag{3}$$

and an observation equation,

$$y_t = g(t, x_t, \omega_t), \tag{4}$$

where $f(\cdot)$ is a known function describing the evolution of the state vector from one time step to the next, $g(\cdot)$ is a known function describing the relation between the observation and the state, and $\varepsilon_t \sim p(\varepsilon_t)$ and $\omega_t \sim p(\omega_t)$ are independent state and observation random errors, respectively, assumed to follow known probability distributions. The state equation in Eq. (3) may then stem from a differential equation describing the behaviour of the system under study. Under this alternative formulation of the state-space model, the distributions $p(x_t|x_{t-1})$ and $p(y_t|x_t)$ in Eqs. (1) and (2) are not specified directly and may not necessarily be known in closed form, but they do follow from Eqs. (3) and (4), respectively. A graphical illustration of the general state-space model is shown in Figure 1.

Two important special cases of the general state-space model are the linear-Gaussian model and the finite state-space HMM. Both of these are central for the work of this thesis. In the linear-Gaussian model, the initial state x_1 is Gaussian, and the state and observation equations in Eqs. (3) and (4) are linear with additive zero-mean Gaussian noise. Specifically, we have

$$x_1 \sim \mathcal{N}(x_1; \mu_1, Q_1),$$

$$x_t = A_t x_{t-1} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(\epsilon_t; 0, Q_t),$$

$$y_t = H_t x_t + \omega_t, \quad \omega_t \sim \mathcal{N}(\omega_t; 0, R_t),$$

or, equivalently,

$$\begin{aligned} p(x_1) &= \mathcal{N}(x_1; \mu_1, Q_1), \\ p(x_t|x_{t-1}) &= \mathcal{N}(x_t; A_t x_{t-1}, Q_t), \\ p(y_t|x_t) &= \mathcal{N}(y_t; H_t x_t, R_t), \end{aligned} \tag{5}$$

where $\mu_1 \in \mathbb{R}^n, Q_t \in \mathbb{R}^{n\times n}$, $A_t \in \mathbb{R}^{n\times n}$, $H_t \in \mathbb{R}^{m\times n}$, $R_t \in \mathbb{R}^{m\times m}$, and $\mathcal{N}(x; \mu, \Sigma)$ denotes the density function of the Gaussian distribution with mean vector μ and covariance matrix Σ. In the finite state-space HMM, x_t is a univariate variable and the state-space $\Omega_x = \{0, 1, \ldots, K-1\}$, $K > 1$ takes on a finite number of distinct values. The forward model $p(x_t|x_{t-1})$ then represents a $K \times K$ transition matrix.

2.2 Filtering, smoothing and prediction

The main objective of state-space modelling is some type of inference about the unobserved state process given the observed sequence of observations. Three common inference procedures are filtering, smoothing and prediction. Suppose in the following that we are currently at time step t. Filtering refers to inference about the *present* state x_t given all past and present observations, $y_{1:t}$. The distribution of interest is then the distribution of x_t given $y_{1:t}$, $p(x_t|y_{1:t})$, called the filtering distribution of x_t. Smoothing refers to inference about a *past* state x_s, $s \leq t$, given the observations $y_{1:t}$, and the distribution of interest is then the smoothing distribution of x_s, $p(x_s|y_{1:t})$. For $s = t$, filtering and smoothing coincides. Finally, prediction refers to inference about *future* states $x_{t+1}, x_{t+2}, \ldots$ given the observations $y_{1:t}$, and the distribution $p(x_{t+k}|y_{1:t})$ for $k \geq 1$ is called the k-step ahead prediction, or forecast, distribution of x_t.

By exploiting conditional independence properties of the state-space representation, the filtering, smoothing and prediction distributions can be computed recursively. Consider first the k-step ahead forecast distribution of x_t, $p(x_{t+k}|y_{1:t})$, and suppose that the filtering distribution of x_t, $p(x_t|y_{1:t})$, is known. Using that x_{t+k} is conditionally independent of $y_{1:t}$ given x_{t+k-1}, the k-step ahead forecast distributions $p(x_{t+k}|y_{1:t})$, $k = 1, 2, \ldots$ can be computed recursively according to

$$p(x_{t+k}|y_{1:t}) = \int_{\Omega_x} p(x_{t+k}|x_{t+k-1}) p(x_{t+k-1}|y_{1:t}) \mathrm{d}x_{t+k-1}. \tag{6}$$

Next, consider the filtering distribution $p(x_t|y_{1:t})$. Using Bayes' rule and that y_t

is conditionally independent of $y_{1:t-1}$ given x_t, the filtering distribution $p(x_t|y_{1:t})$ can be computed recursively for $t = 1, 2, \ldots$ according to

$$p(x_t|y_{1:t}) = \frac{p(x_t|y_{1:t-1})p(y_t|x_t)}{p(y_t|y_{1:t-1})} \tag{7}$$

where

$$p(y_t|y_{1:t-1}) = \int_{\Omega_x} p(x_t|y_{1:t-1})p(y_t|x_t)\mathrm{d}x_t \tag{8}$$

and where the one-step ahead forecast distribution $p(x_t|y_{1:t-1})$ follows from Eq. (6), assuming $p(x_{t-1}|y_{1:t-1})$ is known. This means that the series of filtering distributions can be computed recursively according to a recursion where each iteration consists of two steps: first, a *forecast step*, where the one-step ahead prediction distribution $p(x_t|y_{1:t-1})$ is computed according to Eq. (6), and thereafter, an *update step*, where the filtering distribution $p(x_t|y_{1:t})$ is computed according to Eqs. (7) and (8). In this context, $p(x_t|y_{1:t-1})$ is typically just referred to as the forecast distribution, or simply the prior. The recursion is initialised by setting the forecast distribution of the first iteration equal to the Markov chain initial distribution $p(x_1)$. Finally, suppose T observations, $y_{1:T}$, have been recorded and consider the smoothing distributions $p(x_t|y_{1:T})$ for $t < T$. From the state-space representation it follows that x_t is conditionally independent of $y_{t+1:T}$ given x_{t+1}. That is, we have

$$p(x_t|x_{t+1}, y_{1:T}) = p(x_t|x_{t+1}, y_{1:t}).$$

Using Bayes' rule and that x_{t+1} is conditionally independent of $y_{1:t}$ given x_t, $p(x_t|x_{t+1}, y_{1:t})$ can be expressed as

$$p(x_t|x_{t+1}, y_{1:t}) = \frac{p(x_{t+1}|x_t)p(x_t|y_{1:t})}{p(x_{t+1}|y_{1:t})}.$$

Thereby, if the filtering and one-step ahead prediction distributions have already been calculated for $t = 1$ to T as described above, the smoothing distributions $p(x_t|y_{1:T})$ for $t < T$ can be computed recursively for $t = T-1, T-2, \ldots$ according to

$$p(x_t|y_{1:T}) = p(x_t|y_{1:t}) \int_{\Omega_x} \frac{p(x_{t+1}|y_{1:T})p(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})}\mathrm{d}x_{t+1}. \tag{9}$$

Generally, the integrals in Eqs. (6), (8) and (9) are too complex to be evaluated exactly, and the series of prediction, filtering and smoothing distributions are left intractable. Two exceptions are the linear-Gaussian model and the finite state-space HMM. For the linear-Gaussian model, all distributions involved are Gaussian and the filtering and smoothing recursions lead to the famous Kalman filter and Kalman smoother, respectively (Kalman, 1960). For the finite state-space HMM, the integrals simply represent summations so that no complicated integrals need to be solved. Provided that the number of classes, K, is not too large, it is then possible to perform all the required computations. The forward filtering recursions, followed by the backward smoothing recursions, are then typically referred to as the forward-backward algorithm.

In this thesis, the focus is on the filtering problem. Filtering is in some communities, especially in geophysics, referred to as sequential data assimilation, and the update step of the filtering recursions is often called the analysis step. As mentioned, the solution to the filtering recursions is generally intractable, and approximate strategies are therefore required. In most situations, the best approach is to use an ensemble-based method, where a set of samples, an ensemble, is used to empirically represent the forecast and filtering distributions. A broad range of ensemble-based filtering methods have been proposed, among which we find two main classes: particle filters (Gordon et al., 1993; Doucet et al., 2001; Chopin and Papaspiliopoulos, 2020) and ensemble Kalman filters (EnKFs) (Burgers et al., 1998; Evensen, 2003). While particle filters are based on importance sampling combined with an optional resampling step, EnKFs rely on the assumption of a linear-Gaussian model. Particle filters and the EnKF are described in more detail in the next two sections.

3 Particle filtering

This section gives a brief introduction to the class of ensemble-based filtering methods called particle filters, also referred to as sequential Monte Carlo methods. Particle filters are very general as no assumptions about the distributions of the underlying state-space model are introduced, and they are asymptotically correct in the sense that as the ensemble size goes to infinity, the filters converge to the true filtering solution. Below, we start out in Section 3.1 with a quick review of importance sampling, which represents the foundation of particle filtering methods. Thereafter, we describe importance sampling in a sequential framework in Section 3.2. Finally, we present the most basic particle filtering algorithm, called the bootstrap filter, or the sequential importance resampling (SIR) algorithm, in

Section 3.3.

3.1 Importance sampling

Importance sampling is a Monte Carlo integration technique for estimating properties of a target distribution $p(x)$ while sampling from another distribution. Specifically, consider a distribution $p(x)$,

$$p(x) = \frac{\pi(x)}{\int_{\Omega_x} \pi(z)\mathrm{d}z},$$

where the unnormalised distribution $\pi(x)$ is known in closed form and the normalising constant $\int_{\Omega_x} \pi(x)\mathrm{d}x$ possibly is not. Suppose we are interested in computing the expected value of some function $\zeta(x)$ with respect to $p(x)$,

$$\mathrm{E}_p\left[\zeta(x)\right] = \int_{\Omega_x} \zeta(x)p(x)\mathrm{d}x = \int_{\Omega_x} \zeta(x)\frac{\pi(x)}{\int_{\Omega_x} \pi(z)\mathrm{d}z}\mathrm{d}x, \tag{10}$$

where the subscript p on the left-hand-side is used to express that the expectation is taken over the distribution $p(x)$. Importance sampling represents a method for approximating Eq. (10) when sampling from $p(x)$ is problematic and standard Monte Carlo integration is not an option. Importance sampling relies on the fact that, for some distribution $q(x)$ such that $q(x) > 0$ for all $x \in \Omega_x$ so that $p(x) > 0$, the expectation in Eq. (10) can be rewritten as

$$\mathrm{E}_p\left[\zeta(x)\right] = \int_{\Omega_x} \zeta(x)\frac{\frac{\pi(x)}{q(x)}q(x)}{\int_{\Omega_x} \frac{\pi(z)}{q(z)}q(z)\mathrm{d}z}\mathrm{d}x = \frac{\mathrm{E}_q\left[w(x)\zeta(x)\right]}{\mathrm{E}_q\left[w(x)\right]}$$

where

$$w(x) = \pi(x)/q(x) \tag{11}$$

and the subscript q on the right-hand-side indicates that the expectation is taken with respect to $q(x)$. Usually in this context, $q(x)$ is called the *importance distribution* and $w(x)$ the *importance weight*. If M independent random samples $x^{(1)}, \dots, x^{(M)}$ from $q(x)$ are available, the expectation $\mathrm{E}_p\left[\zeta(x)\right]$ can be approximated as

$$\hat{\mathrm{E}}_p[\zeta(x)] = \frac{\frac{1}{M}\sum_{i=1}^{M} w^{(i)}\zeta(x^{(i)})}{\frac{1}{M}\sum_{j=1}^{M} w^{(j)}} = \sum_{i=1}^{M} \widetilde{w}^{(i)}\zeta\big(x^{(i)}\big), \tag{12}$$

where $w^{(i)} = w\big(x^{(i)}\big)$, and

$$\widetilde{w}^{(i)} = \frac{w^{(i)}}{\sum_{j=1}^{M} w^{(j)}}, \quad i = 1, \dots, M$$

are the *normalised importance weights*. The approximation in Eq. (12) is biased for a finite ensemble size, but unbiased in the limit of an infinite ensemble size (Geweke, 1989). Another important thing to note, is that the weighted ensemble $\{(x^{(i)}, w^{(i)})\}_{i=1}^{M}$ yields an importance sampling approximation to $p(x)$ in the form of a weighted sum of delta masses,

$$\hat{p}(x) = \sum_{i=1}^{M} \widetilde{w}^{(i)} \delta\big(x - x^{(i)}\big),$$

where $\delta\big(x - x^{(i)}\big)$ is the standard dirac delta function.

3.2 Sequential importance sampling

Reconsider hereafter the general state-space model introduced in Section 2. Sequential importance sampling (SIS) uses importance sampling to recursively construct an estimate of the joint posterior distribution $p(x_{1:t}|y_{1:t})$ for $t = 1, 2, \dots$. For each t, an estimate of the filtering distribution $p(x_t|y_{1:t})$ is thereby also obtained. Specifically, SIS involves importance sampling with $p(x_{1:t}|y_{1:t}) \propto p(x_{1:t}, y_{1:t})$ as the target distribution and an importance distribution $q(x_{1:t}|y_{1:t})$ of the form

$$q(x_{1:t}|y_{1:t}) = q(x_{1:t-1}|y_{1:t-1})q(x_t|x_{1:t-1}, y_{1:t}) = q(x_1|y_1) \prod_{j=2}^{t} q(x_j|x_{1:j-1}, y_{1:j}). \quad (13)$$

An importance distribution of this form, combined with the conditional independence properties of the state-space representation, allows us to construct a recursive algorithm. Using Eq. (13) and that the joint distribution $p(x_{1:t}, y_{1:t})$ can be recursively computed according to

$$p(x_{1:t}, y_{1:t}) = p(x_{1:t-1}, y_{1:t-1}) p(y_t|x_t) p(x_t|x_{t-1}),$$

the importance weight in Eq. (11) takes the form

$$\begin{aligned} w\big(x_{1:t}\big) &= \frac{p\big(x_{1:t-1}, y_{1:t-1}\big)p\big(y_t|x_t\big)p\big(x_t|x_{t-1}\big)}{q\big(x_{1:t-1}|y_{1:t-1}\big)q\big(x_t|x_{1:t-1}, y_{1:t}\big)} \\ &= w\big(x_{1:t-1}\big)\frac{p\big(y_t|x_t\big)p\big(x_t|x_{t-1}\big)}{q\big(x_t|x_{1:t-1}, y_{1:t}\big)}. \end{aligned} \tag{14}$$

To initialise the algorithm, we generate an ensemble $\{x_1^{(1)}, \ldots, x_1^{(M)}\}$ of M independent realisations from $q(x_1|y_1)$ and compute the corresponding importance weights $w_1^{(1)}, \ldots, w_1^{(M)}$ according to $w_1^{(i)} = p\big(y_1|x_1^{(i)}\big)p(x_1^{(i)})/q(x_1^{(i)}|y_1)$. Thereafter, for $t = 2, 3, \ldots,$ we sample $x_t^{(i)}$ from $q(x_t|x_{1:t-1}^{(i)}, y_{1:t})$ independently for each $i = 1, \ldots, M$ and compute the associated weights $w_{1:t}^{(i)} = w\big(x_{1:t}^{(i)}\big)$ according to Eq. (14). At each time step t, an empirical estimate of $p(x_{1:t}|y_{1:t})$ follows as

$$\hat{p}(x_{1:t}|y_{1:t}) = \sum_{i=1}^{M} \widetilde{w}_{1:t}^{(i)}\delta\big(x_{1:t} - x_{1:t}^{(i)}\big),$$

where $\widetilde{w}_{1:t}^{(i)}$, $i = 1, \ldots, M$ are the normalised importance weights. The pair $\big(x_{1:t}^{(i)}, w_{1:t}^{(i)}\big)$ is usually called a particle in this context.

An important special case of the SIS framework arises when the Markov forward model $p(x_t|x_{t-1})$ is used as importance distribution $q(x_t|x_{1:t-1}, y_{1:t})$. This is the approach of standard particle filters. The sampling from $p(x_t|x_{t-1})$ can then be interpreted as the forecast step of the filtering recursions, while the consecutive reweighting of the particles corresponds to the update step. Using $p(x_t|x_{t-1})$ as importance distribution, the importance weight function in Eq. (14) simplifies to

$$w(x_{1:t}) = w(x_{1:t-1})p(y_t|x_t),$$

i.e., the updated importance weight at time t is simply obtained by multiplying the previous weight by the likelihood $p(y_t|x_t)$.

3.3 Sequential importance resampling

Although the SIS filter with $p(x_t|x_{t-1})$ as importance distribution yields a consistent estimate of $p(x_{1:t}|y_{1:t})$ for each t, it is well-known that, unless a very high ensemble size is used, the filter tends to collapse after only a few iterations in the sense that almost all of the weight is given to one, or a few, particles, while all the other particles have negligible weight. The *effective* sample size (Kong et al., 1994),

$$M_{\text{eff}} = \frac{1}{\sum_{i=1}^{M} \widetilde{w}_{1:t}^{(i)}},$$

is thereby close to one, and the weighted ensemble $\{(x_{1:t}^{(i)}, w_{1:t}^{(i)})\}_{i=1}^{M}$ is a poor representation of $p(x_{1:t}|y_{1:t})$. A simple approach to prevent filter collapse is to include a resampling step where a new ensemble of state trajectories, $\{\widetilde{x}_{1:t}^{(1)}, \dots, \widetilde{x}_{1:t}^{(M)}\}$, is generated by resampling from $\{x_{1:t}^{(1)}, \dots, x_{1:t}^{(M)}\}$. The most standard resampling approach is to resample the $x_{1:t}^{(i)}$'s by sampling, with replacement, from the set $\{x_{1:t}^{(1)}, \dots, x_{1:t}^{(M)}\}$ with probabilities according to the corresponding normalised importance weights. This results in a filter known as the bootstrap filter (Gordon et al., 1993), or the sequential importance resampling (SIR) algorithm. More advanced resampling techniques, such as residual resampling (Lui and Chen, 1998) and stratified resampling (Carpenter et al., 1999), have also been proposed. After the resampling step, the $\widetilde{x}_{1:t}^{(i)}$'s are assigned equal weights, so that our new weighted ensemble is $\{(\widetilde{x}_{1:t}^{(i)}, \frac{1}{M})\}_{i=1}^{M}$. An approximation to the joint posterior distribution $p(x_{1:t}|y_{1:t})$ then follows as

$$\hat{p}(x_{1:t}|y_{1:t}) = \sum_{i=1}^{M} \frac{1}{M} \delta\big(x_{1:t} - \widetilde{x}_{1:t}^{(i)}\big).$$

If interest is only in the filtering distributions $p(x_t|y_{1:t})$, only $x_t^{(i)}$ needs to be resampled and not the entire trajectory $x_{1:t}^{(i)}$, since a consequence of the resampling is that $w(x_{1:t})$ is no longer dependent on $x_{1:t-1}$.

Although particle filters may appear to solve the smoothing problem for free since an importance sampling approximation to the joint posterior distribution $p(x_{1:t}|y_{1:t})$ is obtained at every time step, particle filters are generally not used to solve the smoohting problem. For $s \ll t$, the estimate of the marginal distribution $p(x_s|y_{1:t})$ obtained from the estimate of the joint distribution $p(x_{1:t}|y_{1:t})$ is generally rather poor. If interest is in the smoothing problem, it is better to first run a particle filter in the usual way, and thereafter do a backwards sweep and update the trajectories with a particle smoother (Doucet et al., 2000).

The resampling step can in some situations prevent the basic SIS filter from collapsing. However, it is generally not sufficient. It can be shown that the number of particles required to avoid filter collapse, when $p(x_t|x_{t-1})$ is used as importance distribution and no resampling is performed, grows exponentially with the dimension of the state vector (Snyder et al., 2008), and it is unlikely that a simple resampling can defeat this so-called curse of dimensionality. As pointed out in Doucet et al. (2000), resampling can even be harmful as it introduces other theoretical and practical issues. In their simplest form, particle filters are therefore, generally, not suited for high-dimensional filtering problems. There is, however, a lot of ongoing research in this area, and more advanced schemes are in development. A nice review can be found in van Leeuwen et al. (2019).

4 The ensemble Kalman filter

This section describes the ensemble Kalman filter (EnKF), an ensemble-based version of the traditional Kalman filter. Many variations of the original EnKF scheme, as presented in Burgers et al. (1998), have been proposed in the literature. The many contributions can be divided into two main categories, stochastic filters and deterministic filters, the latter also known as ensemble square root filters (EnSRFs). For simplicity, we restrict in the following the attention to linear-Gaussian likelihood models as in Eq. (5). It is, however, possible to modify the EnKF to allow for non-Gaussian, non-linear likelihood models, although this complicates the theoretical justification of what is really going on in the filter.

4.1 Kalman filter

For the linear-Gaussian state-space model introduced in Section 2.1, the filtering recursions lead to the well-known Kalman filter. The series of forecast and filtering distributions are in this case all Gaussian,

$$p(x_t|y_{1:t}) = \mathcal{N}(x_t; \widetilde{\mu}_t, \widetilde{P}_t),$$

and

$$p(x_{t+1}|y_{1:t}) = \mathcal{N}(x_{t+1}; \mu_{t+1}, P_{t+1}),$$

and the Kalman filter provides recursive formulas for the associated parameters $\widetilde{\mu}_t, \widetilde{P}_t$, μ_{t+1}, and P_{t+1}. Specifically, starting from the initial distribution $p(x_1)$ which, by assumption, is Gaussian with known mean vector μ_1 and covariance matrix Q_1, and setting $P_1 = Q_1$, the Kalman filter computes the filtering parameters $\widetilde{\mu}_t$, $\widetilde{P}_t$ and the forecast parameters μ_{t+1}, P_{t+1} recursively for $t = 1, 2, \ldots$ according to

$$\widetilde{\mu}_t = \mu_t + K_t(y_t - H_t\mu_t), \tag{15}$$

$$\widetilde{P}_t = (I_n - K_t H_t) P_t, \tag{16}$$

$$\mu_{t+1} = A_t \widetilde{\mu}_t,$$

and

$$P_{t+1} = A_t \widetilde{P}_t A_t^\top + Q_t,$$

where I_n is the $n \times n$ identity matrix, and

$$K_t = P_t H_t^\top \left(H_t P_t H_t^\top + R_t \right)^{-1} \tag{17}$$

is the so-called Kalman gain matrix.

4.2 The stochastic EnKF

The stochastic EnKF was first introduced in the geophysics literature as an alternative to the traditional Kalman filter. Geophysical models, such as reservoir models or models of the atmosphere, preclude straightforward implementation of the traditional Kalman filter for two main reasons. Firstly, they are typically of such high dimensions that explicit storage and computation of the full $n \times n$ Kalman filter covariance matrices become problematic. Secondly, they usually involve a forward model $p(x_t|x_{t-1})$ which is non-linear and/or non-Gaussian. By exploiting the ensemble representations of the forecast and filtering distributions, the EnKF avoids explicit computation of the full $n \times n$ covariance matrices and is able to cope with mild features of non-linearity and non-Gaussianity in the forward model.

Like the traditional Kalman filter, each iteration of the EnKF involves a forecast step and an update step. The update step can be viewed as an ensemble-based version of the update step of the traditional Kalman filter. Iteration number t starts with the assumption that a forecast ensemble, $\{x_t^{(1)}, \ldots, x_t^{(M)}\}$, with independent realisations from the forecast distribution $p(x_t|y_{1:t-1})$ is available. In reality, this assumption holds only approximately. The update step is then performed by first (implicitly) approximating the forecast model $p(x_t|y_{1:t-1})$ with a Gaussian distribution $\mathcal{N}(x_t; \hat{\mu}_t, \hat{P}_t)$ with mean vector $\hat{\mu}_t$ equal to the sample mean of the forecast ensemble,

$$\hat{\mu}_t = \frac{1}{M} \sum_{i=1}^{M} x_t^{(i)},$$

and covariance matrix $\hat{P}_t$ equal to the sample covariance,

$$\hat{P}_t = X_t X_t^\top, \tag{18}$$

where

$$X_t = \frac{1}{\sqrt{M-1}} \left(x_t^{(1)} - \hat{\mu}_t, \dots, x_t^{(M)} - \hat{\mu}_t \right)$$

is the so-called forecast ensemble-anomaly matrix. Thereafter, each forecast sample is linearly shifted according to

$$\widetilde{x}_t^{(i)} = x_t^{(i)} + \hat{K}_t \big(y_t - H_t x_t^{(i)} + \omega_t^{(i)} \big), \tag{19}$$

where $\omega_t^{(i)} \sim N(y_t; 0, R_t)$, $\omega_t^{(1)}, \dots, \omega_t^{(M)}$ are all independent, and

$$\hat{K}_t = X_t (H_t X_t)^\top \left(H_t X_t (H_t X_t)^\top + R_t \right)^{-1} \tag{20}$$

is the Kalman gain matrix in Eq. (17), only P_t is replaced with the sample covariance matrix $\hat{P}_t$ in Eq. (18). The justification of the update in Eq. (19) is that, under the assumption that the Gaussian approximation $\mathcal{N}(x_t; \hat{\mu}_t, \hat{P}_t)$ is the correct forecast model, i.e. that $x_t^{(1)}, \dots, x_t^{(M)}$ are independent samples from $\mathcal{N}(x_t; \hat{\mu}_t, \hat{P}_t)$, the update yields independent samples $\widetilde{x}_t^{(1)}, \dots, \widetilde{x}_t^{(M)}$ from the corresponding correct filtering distribution, which then is a Gaussian $\mathcal{N}(x_t; \hat{\hat{\mu}}_t, \hat{\hat{P}}_t)$ with mean $\hat{\hat{\mu}}_t$ and covariance $\hat{\hat{P}}_t$ available from the Kalman filter equations in Eqs. (15) and (16),

$$\hat{\hat{\mu}}_t = \hat{\mu}_t + \hat{K}_t (\hat{\mu}_t - H_t \hat{\mu}_t) \tag{21}$$

and

$$\hat{\hat{P}}_t = (I_n - \hat{K}_t H_t) \hat{P}_t. \tag{22}$$

In geophysical applications, the ensemble size M is typically much smaller than the state dimension n due to computer demanding forward models $p(x_t | x_{t-1})$. The observation dimension, m, is also smaller than n, though considerably bigger than M. From Eqs. (19) and (20), one should note that explicit storage and computation of the full $n \times n$ covariance matrix $\hat{P}_t$ can be avoided and that the largest matrices that need to be maintained are of size $n \times M$, $n \times m$ and $m \times m$. This allows for an efficient numerical implementation compared to the traditional Kalman filter and makes the EnKF computationally feasible also in large-scale applications.

Having generated the filtering ensemble $\{\widetilde{x}_t^{(1)}, \dots, \widetilde{x}_t^{(M)}\}$ according to Eq. (19), the forecast step is performed by generating a new forecast ensemble, $\{x_{t+1}^{(1)}, \dots, x_{t+1}^{(M)}\}$, by simulating from the Markov forward model,

$$x_{t+1}^{(i)} | \widetilde{x}_t^{(i)} \sim p\big(x_{t+1} | \widetilde{x}_t^{(i)}\big), \tag{23}$$

independently for $i = 1, \dots, M$. In contrast to the update step, which relies on Gaussian approximations, the forecast step is exact in the sense that under the assumption that $\widetilde{x}_t^{(1)}, \dots, \widetilde{x}_t^{(M)}$ are exact and independent samples from the true filtering distribution $p(x_t | y_{1:t})$, Eq. (23) returns forecast samples $x_{t+1}^{(1)}, \dots, x_{t+1}^{(M)}$ that are exact and independent samples from the true forecast distribution $p(x_{t+1} | y_{1:t})$.

If the underlying state-space model really is linear-Gaussian, the EnKF is correct in the limit of an infinite ensemble size. The solution it then provides corresponds to that of the Kalman filter. In the more general case of a non-linear, non-Gaussian model, the filter provides a biased solution. However, if there are non-Gaussian features present in the forecast ensemble, it is to some extent possible for the filtering ensemble to capture some of these properties, since it is obtained by linearly shifting the forecast samples.

4.3 EnSRFs

EnSRFs perform a deterministic version of the update in Eq. (19). As the traditional, stochastic EnKF, EnSRFs start by replacing the forecast distribution $p(x_t | y_{1:t-1})$ with a Gaussian approximation $\mathcal{N}(x_t; \hat{\mu}_t, \hat{P}_t)$ from which a corresponding Gaussian approximation $\mathcal{N}(x_t; \hat{\tilde{\mu}}_t, \hat{\tilde{P}}_t)$ to the filtering distribution $p(x_t | y_{1:t})$ follows from Bayes' rule. While the stochastic EnKF updates the forecast ensemble so that the sample mean and sample covariance of the updated ensemble equal $\hat{\tilde{\mu}}_t$ and $\hat{\tilde{P}}_t$ in expectation, or in the limit of an infinite ensemble size, EnSRFs deterministically update the ensemble so that the sample mean and sample covariance equal $\hat{\tilde{\mu}}_t$ and $\hat{\tilde{P}}_t$ *exactly.*

EnSRFs involve computations with matrix square roots. Similarly to the ensemble-anomaly matrix X_t of the forecast ensemble, an ensemble-anomaly matrix $\widetilde{X}_t$ of the filtering ensemble can be defined as

$$\widetilde{X}_t = \frac{1}{\sqrt{M-1}} \left(\Delta\widetilde{x}_t^{(1)}, \dots, \Delta\widetilde{x}_t^{(M)} \right), \tag{24}$$

where

$$\Delta\widetilde{x}_t^{(i)} = \widetilde{x}_t^{(i)} - \frac{1}{M}\sum_{j=1}^{M}\widetilde{x}_t^{(j)}. \tag{25}$$

The ensemble-anomaly matrices X_t and $\widetilde{X}_t$ are matrix square roots of the sample covariance matrices of the forecast and filtering ensembles, respectively. The strategy of EnSRFs is to compute an ensemble-anomaly matrix $\widetilde{X}_t$ by requiring that the sample covariance matrix of the filtering ensemble is exactly equal to $\hat{\tilde{P}}_t$. Specifically, this requirement entails that

$$\hat{\tilde{P}}_t = \widetilde{X}_t\widetilde{X}_t^\top. \tag{26}$$

Using Eqs. (18) and (26), and defining $D_t = H_t X_t X_t^\top H_t^\top + R_t$, Eq. (22) can be written as

$$\widetilde{X}_t\widetilde{X}_t^\top = \left(I_n - X_t X_t^\top H_t^\top D_t^{-1} H_t\right) X_t X_t^\top.$$

Rearranging terms on the right-hand-side, we obtain

$$\widetilde{X}_t\widetilde{X}_t^\top = X_t\left(I_M - X_t^\top H_t^\top D_t^{-1} H_t X_t\right) X_t^\top.$$

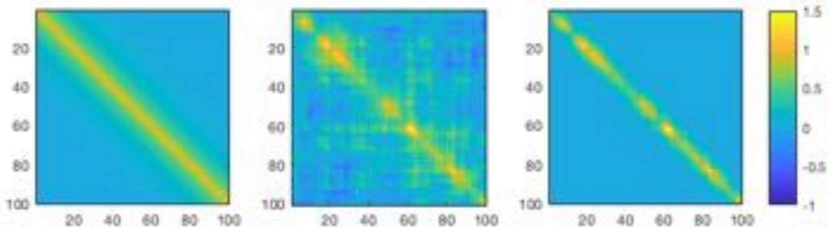
Thereby, we see that a posterior ensemble-anomaly matrix $\widetilde{X}_t$ with the desired properties can be obtained as

$$\widetilde{X}_t = X_t W_t U$$

where $W_t \in \mathbb{R}^{M\times M}$ is a matrix square root of $\left(I_M - X_t^\top H_t^\top D_t^{-1} H_t X_t\right)$, i.e.

$$W_t W_t^\top = \left(I_M - X_t^\top H_t^\top D_t^{-1} H_t X_t\right),$$

and U is an $M \times M$ orthonormal matrix, i.e. $UU^\top = U^\top U = I_M$. The posterior ensemble members $\widetilde{x}_t^{(1)},\ldots,\widetilde{x}_t^{(M)}$ can thereafter be obtained by inserting $\frac{1}{M}\sum_{j=1}^{M}\widetilde{x}_t^{(j)} = \hat{\tilde{\mu}}_t$ in Eq. (25). Since the matrices W_t and U are not unique, except in the univariate case, a variety of EnSRF schemes can be formulated. As such, several EnSRFs have been proposed in the literature, e.g. Anderson (2001), Bishop et al. (2001), Whitaker and Hamill (2002), and Evensen (2004).



(a) (b) (c)

Figure 2: Demonstration of covariance localisation. (a) True covariance matrix, P. (b) Sample covariance matrix, $\hat{P}$, obtained from $M = 20$ random samples. (c) Regularised covariance matrix obtained from the Schur product of ρ and $\hat{P}$ with ρ given by Eq. (29) using $L = 10$.

4.4 Localisation and inflation

The use of a finite ensemble size, and especially the use of an ensemble size much smaller than the state dimension, comes at a price. When $M \ll n$, the forecast sample covariance matrix $\hat{P}_t$, whose rank is at most $M-1$, is severely rank deficient and usually a poor substitute for the true, possibly full rank, covariance matrix P_t. In particular, $\hat{P}_t$ is known to suffer from what is known as *spurious correlations*, which refers to overestimation of off-diagonal elements of P_t that are supposed to be close to zero. This undesirable behaviour is demonstrated in Figure 2(b) which shows the sample covariance matrix obtained from $M = 20$ independent samples from a Gaussian distribution of dimension $n = 100$ with zero mean and covariance matrix shown in Figure 2(a). Spurious correlations can lead to an updated ensemble with a too small spread which, when done sequentially, can result in filter divergence in the sense that the variability of the forecast and filtering ensembles become smaller and smaller and the ensemble mean eventually drifts away from the truth. Spurious correlations are a natural result of sampling errors and occur also for $M > n$, however when M is sufficiently large the effect is often small enough to avoid filter divergence. Two techniques proposed to correct for spurious correlations are localisation and inflation. Both techniques are commonly applied in practical applications, often in combination.

Localisation

Localisation relies on the fact that, in most spatial geophysical systems, correlations between variables decrease rapidly with the distance between them, often exponentially. The correlation between two variables x_t^i and x_t^j of x_t should therefore be close to zero if the indices i and j correspond to locations far apart in space. Two different localisation techniques have been proposed in the literature: covariance localisation and domain localisation. A nice review of the two procedures can be found in Sakov and Bertino (2011), where also the relation between them is investigated.

Covariance localisation (Hamill and Whitaker, 2001; Houtekammer and Mitchell, 2001) seeks to increase the rank of the estimated forecast covariance matrix and to suppress spurious correlations by replacing the sample covariance matrix with its Schur (element-wise) product with some well chosen correlation matrix $\rho \in \mathbb{R}^{n \times n}$,

$$\hat{P}_t \to \rho \circ \hat{P}_t. \tag{27}$$

The matrix ρ is chosen so that it reflects how the correlations between variables decrease with the distance between them as seen in real geophysical systems. The Schur product in Eq. (27) should then result in a regularised covariance matrix where the spurious correlations are dampened. A common approach is to use the Gaspari-Cohn function (Gaspari and Cohn, 1999),

$$G(r) = \begin{cases} 1 - \frac{5}{3}r^2 + \frac{5}{8}|r|^3 + \frac{1}{2}r^4 - \frac{1}{4}|r|^5, & \text{if } 0 \leq |r| < 1, \\ 4 - 5|r| + \frac{5}{3}r^2 + \frac{5}{8}|r|^3 - \frac{1}{2}r^4 + \frac{1}{12}|r|^5 - \frac{2}{3|r|}, & \text{if } 1 \leq |r| < 2, \\ 0, & \text{if } |r| \geq 2, \end{cases} \tag{28}$$

and define the entries of ρ as

$$\rho_{ij} = G((i-j)/L), \tag{29}$$

where L is a so-called correlation length which determines the rate at which the correlations decrease towards zero. Figure 2(c) shows the covariance matrix obtained from the Schur product of the sample covariance matrix shown in Figure 2(b) and a correlation matrix ρ defined by Eq. (29) with $L = 10$. The downside of covariance localisation is that, brute force, it involves storage and computation of $n \times n$ matrices. A possible way to circumvent this, is to choose ρ as sparse.

Domain localisation (Ott et al., 2004; Hunt et al., 2007; Janjic et al., 2011), or local analysis, instead divides the state vector into several disjoint subsets and performs a 'local' update for each subset. In each of the updates, only a corresponding local subset of the observation vector, containing observations within some chosen cut-off radius from the centre of the assimilation region, is considered. Computationally, this approach is advantageous over covariance localisation as it can exploit the ensemble representations and avoid explicit maintenance of $n \times n$ matrices. An issue of concern, however, is lack of smoothness in the updated realisations due to the division of the global domain into several subdomains. Different techniques have been proposed to correct for this issue. For example, Hunt et al. (2007) propose to weight the observation error covariance matrix so that observations further away from the assimilation region are assigned larger variances.

Inflation

For high-dimensional models, localisation alone is often not sufficient to avoid filter divergence, and covariance inflation (Anderson and Anderson, 1999) is also applied to stabilise the filter. With inflation, the estimated forecast distribution is artificially broadened by multiplying the sample covariance matrix $\hat{P}_t$ by a factor $\lambda > 1$,

$$\hat{P}_t \to \lambda \hat{P}_t,$$

or, equivalently, by multiplying the ensemble-anomaly matrix X_t by a factor $\sqrt{\lambda}$, i.e. $X_t \to \sqrt{\lambda} X_t$. The inflation factor λ is usually only slightly larger than one and needs to be tuned to obtain satisfactory performance. Such tuning can be a burden computationally, but adaptive schemes have been proposed (e.g., Wang and Bishop, 2003; Anderson, 2007, 2009).

An illustrative example

Here, we present a simple simulation example which illustrates the potential effect of using covariance localisation in the EnKF. The example involves a linear-Gaussian state-space model, and results obtained using the stochastic EnKF scheme, with and without covariance localisation, are compared. For demonstration purposes, we also present output from a modified EnKF where the true forecast covariance matrices, computed with the Kalman filter, are used to update the ensemble in each iteration. Of course, such an approach is not something one would be able to run in practice, but for the purpose of this experiment it is convenient to use the output as a reference, since it reflects how an ensemble of M realisations ideally should look like. The dimension of the state vector x_t is $n = 100$, and for every fifth variable of x_t there is an observation y_t^j, so that the dimension of the observation vector y_t is $m = 20$. The relatively small ensemble size $M = 20$ is used in all three schemes, and the correlation matrix ρ used in the localisation procedure is defined by Eq. (29) with $L = 10$.

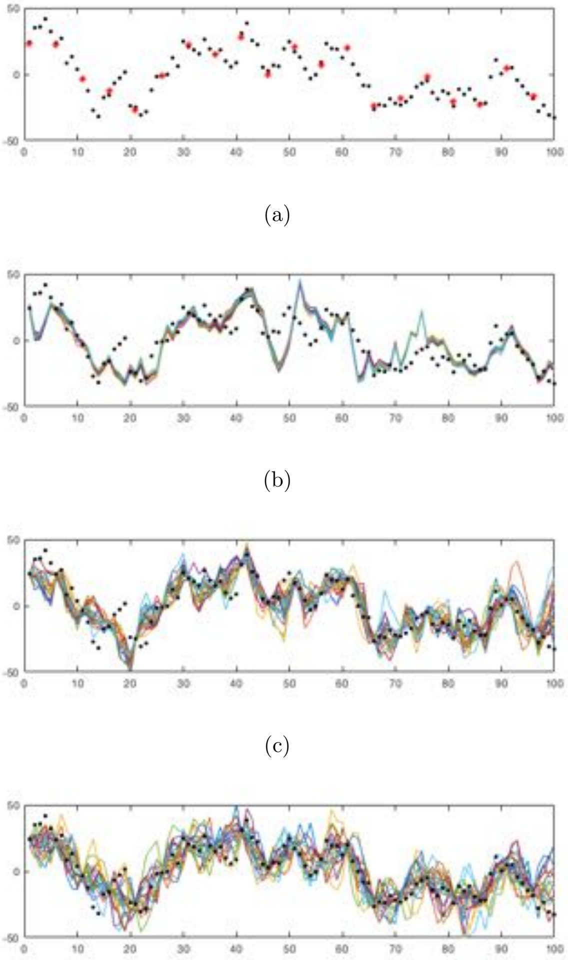
Figure 3(a) shows the true state vector x_t and the observation vector y_t at time step $t = 40$, and Figures 3(b)-(d) show the posterior ensemble members $\widetilde{x}_t^{(1)}, \ldots, \widetilde{x}_t^{(M)}$ obtained from the three different EnKF schemes described above at this time step. Notice in particular that the ensemble spread in Figure 3(b) obtained with the traditional EnKF, with no localisation, unmistakably is much too narrow compared to the spread in Figure 3(d) obtained with the scheme using the correct covariance matrices. The true values of the x_t^j's are then also quite often far outside the spread of the ensemble. In Figure 3(c), which shows the results from the covariance localisation scheme, this effect is considerably reduced, and the ensemble spread is more comparable to that in Figure 3(d).

5 Summary of papers

The ultimate goal of my Ph.D. was to generalise the EnKF and to use this generalised scheme to develop an ensemble-based filtering method, or essentially an ensemble *updating* method, for high-dimensional, categorical state vectors. The reason for wanting to generalise the EnKF is that many studies show that the filter provides quite good results even in non-linear, non-Gaussian situations, so therefore it would be interesting to investigate whether some of the underlying properties of the filter that contribute to this appealing behaviour can be transferred to a categorical sample space. Throughout the thesis, we focus on state vectors with a one-dimensional spatial arrangement, meaning that the vector is spatially arranged along a line. Extending the proposed methods to two, and possibly three, dimensions is an interesting area for future research. The remains of the thesis consists of four papers, all closely related. The papers can be read independently, but we recommend reading paper I before the others, especially before papers II and III. Below, we briefly summarise each paper.

Paper I

The first paper, "Ensemble updating of binary state vectors by maximising the expected number of unchanged components", describes our first effort on



(a)

(b)

(c)

(d)

Figure 3: EnKF simulation example: (a) True state vector x_t (black dots) and observed vector y_t (red stars) at time $t = 40$. (b) True state vector x_t (black dots) and filtering ensemble members (coloured lines) at time $t = 40$ obtained using the standard stochastic EnKF. (c) Corresponding output as in (b), obtained using stochastic EnKF with covariance localisation, where ρ is defined by Eq. (29) with $L = 10$. (d) Corresponding output as in (b) and (c), obtained using a modified stochastic EnKF where the true Kalman filter forecast covariance matrices are used in the ensemble update.

developing a new ensemble updating method for categorical state vectors. As implied by the title, the paper restricts the focus to vectors where each element is a binary variable. The paper starts with the description of a general ensemble updating framework based on a generalisation of the statistical properties of the EnKF. In the EnKF, a Gaussian approximation to the forecast distribution is (implicitly) constructed. Combined with the assumption of a linear-Gaussian likelihood model, a corresponding Gaussian approximation to the filtering distribution can be computed according to Bayes' rule. Given that the Gaussian approximation to the forecast distribution is correct, the EnKF linear update corresponds to conditional simulation from a Gaussian distribution such that the marginal distribution of each updated sample is equal to the Gaussian approximation to the filtering distribution. More generally, one can imagine to proceed in a similar fashion, but pursue another parametric model than the Gaussian. That is, instead of assuming that the forecast model is Gaussian and that the likelihood model is linear-Gaussian, other models can be chosen. Moreover, instead of linearly shifting the forecast samples, the posterior samples can be obtained by conditional simulation from some distribution such that, under the assumption that the forecast samples are distributed according to the assumed forecast model, the marginal distribution of each updated sample is equal to the corresponding assumed filtering distribution. Generally, an infinite number of such conditional distributions may exist. To choose a solution, one could for example seek a solution which is optimal with respect to some chosen optimality criterion.

To update a vector of binary variables, we propose to construct a first-order Markov chain approximation to the forecast distribution and assume that the elements of the observation vector are conditionally independent. This choice of forecast and likelihood models constitutes an HMM and returns a corresponding tractable first-order Markov chain approximation to the filtering distribution. Based on the assumed HMM, the next task is to construct an appropriate conditional distribution from which the posterior samples can be simulated. Because of the discrete context, this conditional distribution is a transition matrix, not a density as in the EnKF. A simple yet naïve option is to set the transition matrix equal to the already established first-order Markov chain approximation to the filtering distribution. However, this naïve approach entails that the posterior samples are simulated independently of the forecast samples and may result in that important information about the *true* forecast and filtering models, possibly not captured with the assumed Markov chain models, is lost. To capture more information from the forecast samples, we propose to construct an optimal transition matrix with respect to the optimality criterion of maximising the expected number of components of a forecast realisation that remain unchanged. A dynamic programming algorithm for recursively constructing the optimal solution is presented, and the proposed procedure is demonstrated in a geophysics-inspired simulation example.

Paper II

The second paper, "Geophysics-based fluid-facies predictions based on ensemble updating of binary state vectors", is much more applied than the remaining papers. Basically, the paper presents a synthetic geophysical filtering problem where the method proposed in Paper I is applied. The problem considered is a two-phase fluid flow problem originating from water injection in a petroleum reservoir. Based on noisy measurements of a geophysical property called resistivity recorded at various times and at different locations in the reservoir, the goal is to monitor the oil displacement. Simulation examples with a two-dimensional reservoir model are presented. Here, to deal with the two-dimensional context, the updating of the variables associated with each column of the grid is done independently of the remaining variables.

Paper III

The third paper, "A generalised and fully Bayesian ensemble updating framework", is an extension of the work presented in Paper I. The contribution of the paper is three-fold. Firstly, the general updating framework proposed in Paper I is modified to a Bayesian context where the parameters of the assumed forecast distribution are also treated as random variables. Secondly, the proposed Bayesian framework is investigated under the assumption of a linear-Gaussian model. An important result of this part of the paper is the proof that a particular EnSRF scheme is optimal with respect to the optimality criterion of minimising the expected Mahalanobis distance between a prior and posterior ensemble member. Thirdly, the framework is examined under the assumption of a binary HMM. Simulation examples for both the linear-Gaussian model and the binary HMM are presented.

A consequence of the proposed Bayesian setup is that, prior to the updating of each forecast ensemble member, a corresponding parameter vector needs to be simulated, and this simulation is to be done conditionally on both the incoming observation and all the forecast samples except the forecast sample which is to be updated. This is different from existing fully Bayesian approaches, such as the hierarchical EnKF (HEnKF) of Myrseth and Omre (2010), where the parameters are simulated conditionally on *all* the forecast samples (including the sample which is to be updated), but not the data. In the simulation example with the linear-Gaussian model, we observe that the exclusion of the forecast sample which is to be updated can have a major impact on the results when the ensemble size is small. In particular, compared to the traditional EnKF and the HEnKF, we observe that the proposed Bayesian EnSRF scheme provides more reliable results and gives a much better representation of the uncertainty.

Currently, although referred to as a paper in the thesis, Paper III is not really a paper, but a technical report. However, it will be submitted to a journal in the future, after some revisions.

Paper IV

The fourth paper, "Ensemble updating of categorical state vectors", is another extension of work presented in Paper I. The paper follows in the same Bayesian spirit as Paper III, but focuses entirely on categorical state vectors. A slightly modified version of the general Bayesian framework proposed in Paper III is presented and an improved version of the updating method proposed in Paper I is described. Two important limitations about the algorithm proposed in Paper I are that it works for binary variables only and that the assumed forecast distribution is restricted to be a first-order Markov chain. In Paper IV, we address these two issues and present an improved method which is computationally feasible also for situations with more than two classes and which allows for a higher-order Markov chain as the assumed forecast distribution. While the algorithm proposed in Paper I is based on a certain directed acyclic graph for the dependencies between the variables of a prior and posterior ensemble member, the algorithm proposed in Paper IV instead makes use of an undirected graph. The chosen structure of this undirected graph makes it possible to efficiently construct the optimal transition matrix by solving a linear program. A simulation example where each variable of the state vector can take three different values is presented.

References

Anderson, J. L. (2001). An ensemble adjustment Kalman filter for data assimilation. *Monthly Weather Review*, *129*, 2884–2903.

Anderson, J. L. (2007). An adaptive covariance inflation error correction algorithm for ensemble filters. *Tellus A: Dynamic Meteorology and Oceanography*, *59*, 210–224.

Anderson, J. L. (2009). Spatially and temporally varying adaptive covariance inflation for ensemble filters. *Tellus A: Dynamic Meteorology and Oceanography*, *61*, 72–83.

Anderson, J. L., & Anderson, S. L. (1999). A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, *127*, 2741–2758.

Bishop, C. H., Etherton, B. J., & Majumdar, S. J. (2001). Adaptive sampling with the ensemble transform Kalman filter. Part 1: Theoretical aspects. *Monthly Weather Review*, *129*, 420–436.

Burgers, G., van Leeuwen, P. J., & Evensen, G. (1998). Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, *126*, 1719–1724.

Carpenter, J., Clifford, P., & Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings - Radar, Sonar and Navigation*, *146*, 2–7.

Chopin, N., & Papaspiliopoulos, O. (2020). *An introduction to sequential Monte Carlo*. Springer-Verlag, Cham.

Doucet, A., de Freitas, N., & Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. Springer-Verlag, New York.

Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, *10*, 197–208.

Evensen, G. (2003). The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, *53*, 343–367.

Evensen, G. (2004). Sampling strategies and square root analysis schemes for the EnKF. *Ocean Dynamics*, *54*, 539–560.

Gaspari, G., & Cohn, S. E. (1999). Construction of correlation functions in two or three dimensions. *Quarterly Journal of the Royal Meteorological Society*, *125*, 723–757.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, *57*, 1317–1339.

Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE-Proceedings-F*, *140*, 107–113.

Hamill, T. M., & Whitaker, J. S. (2001). Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Monthly Weather Review*, *129*, 2776–2790.

Houtekammer, P. L., & Mitchell, H. L. (2001). A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, *129*, 123–137.

Hunt, B. R., Kostelich, E. J., & Szunyogh, I. (2007). Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena*, *230*, 112–126.

Janjic, T., Nerger, L., Albertella, A., Schröter, J., & Skachko, S. (2011). On domain localization in ensemble-based Kalman filter algorithms. *Monthly Weather Review*, *139*, 2046–2060.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, *82*, 35–45.

Kong, A., Liu, J. S., & Wong, W. H. (1994). Sequential imputation and Bayesian missing data problems. *Journal of the American Statistical Association*, *89*, 278–288.

Künsch, H. R. (2000). State space and hidden Markov models. In O. E. Barndorff-Nielsen, D. R. Cox, & C. Klüppelberg (Eds.), *Complex stochastic systems*. Chapman and Hall/CRC, Chap. 3, p. 109-174.

Lui, J. S., & Chen, R. (1998). Sequential Monte-Carlo methods for dynamical systems. *Journal of the American Statistical Association*, *90*, 567–576.

Myrseth, I., & Omre, H. (2010). Hierarchical ensemble Kalman filter. *SPE Journal*, *15*, 569–580.

Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., Kalnay, E., Patil, D. J., & Yorke, J. A. (2004). A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, *56*, 415–428.

Sakov, P., & Bertino, L. (2011). Relation between two common localisation methods for the EnKF. *Computational Geosciences*, *15*, 225–237.

Snyder, C., Bengtsson, T., Bickel, P., & Anderson, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, *136*, 4629–4640.

van Leeuwen, P. J., Künsch, H. R., Nerger, L., Potthast, R., & Reich, S. (2019). Particle filters for high-dimensional geoscience applications: A review. *Quarterly Journal of the Royal Meteorological Society*, *145*, 2335–2365.

Wang, X., & Bishop, C. H. (2003). A comparison of breeding and ensemble transform Kalman filter ensemble forecast schemes. *Journal of the Atmospheric Sciences*, *60*, 1140–1158.

Whitaker, J. S., & Hamill, T. M. (2002). Ensemble data assimilation without perturbed observations. *Monthly Weather Review*, *130*, 1913–1924.

Paper I

Ensemble updating of binary state vectors by maximising the expected number of unchanged components

Margrethe Kvale Loe and Håkon Tjelmeland

To appear in *Scandinavian Journal of Statistics*

Received: 11 February 2019 | Revised: 4 March 2020 | Accepted: 24 June 2020

DOI: 10.1111/sjos.12483

ORIGINAL ARTICLE

Scandinavian Journal of Statistics

Ensemble updating of binary state vectors by maximizing the expected number of unchanged components

Margrethe Kvale Loe | **Håkon Tjelmeland**

Department of Mathematical Sciences, Norwegian University of Science and Technology, Norway

Correspondence
Margrethe K. Loe, Alfred Getz' vei 1, Gløshaugen, 7034 Trondheim, Norway.
Email: margrethe.loe@ntnu.no

Abstract

The main challenge in ensemble-based filtering methods is the updating of a prior ensemble to a posterior ensemble. In the ensemble Kalman filter (EnKF), a linear-Gaussian model is introduced to overcome this issue, and the prior ensemble is updated with a linear shift. In the current article, we consider how the underlying ideas of the EnKF can be applied when the state vector consists of binary variables. While the EnKF relies on Gaussian approximations, we instead introduce a first-order Markov chain approximation. To update the prior ensemble we simulate samples from a distribution which maximizes the expected number of equal components in a prior and posterior state vector. The proposed approach is demonstrated in a simulation experiment where, compared with a more naive updating procedure, we find that it leads to an almost 50% reduction in the difference between true and estimated marginal filtering probabilities with respect to the Frobenius norm.

KEYWORDS

data assimilation, ensemble Kalman filter, hidden Markov models

1 | INTRODUCTION

A state-space model consists of a latent $\{x^t\}_{t=1}^{\infty}$ process and an observed $\{y^t\}_{t=1}^{\infty}$ process, where y^t is a partial observation of x^t. More specifically, the y^t's are assumed to be conditionally independent given the x^t process and y^t only depends on x^t. Estimation of the latent variable at time t, x^t, given all observations up to this time, $y^{1:t} = (y^1, \dots , y^t)$, is known as the filtering or data assimilation problem. In the linear Gaussian situation an easy to compute and exact solution is available by the famous Kalman filter. In most nonlinear or non-Gaussian situations, however, no computationally feasible exact solution exists and ensemble methods are therefore frequently adopted. The distribution $p(x^t|y^{1:t})$ is then not analytically available, but is represented by a set of realizations $\tilde{x}^{t(1)}, \dots , \tilde{x}^{t(M)}$ from this filtering distribution. Assuming such an ensemble of realizations to be available for time $t-1$, the filtering problem is solved for time t in two steps. First, based on the Markov chain model for the x^t process, each $\tilde{x}^{t-1(i)}$ is used to simulate a corresponding forecast realization $x^{t(i)}$, which marginally are independent samples from $p(x^t|y^{1:t-1})$. This is known as the forecast or prediction step. Second, an update step is performed, where each $x^{t(i)}$ is updated to take into account the new observation y^t and the result is an updated ensemble $\tilde{x}^{t(1)}, \dots , \tilde{x}^{t(M)}$ which represents the filtering distribution at time t, $p(x^t|y^{1:t})$. The updating step is the difficult one and the different strategies that have been proposed can be classified into two classes, particle filters and ensemble Kalman filters.

In particle filters (Doucet, de Freitas, & Gordon, 2001) each filtering realization $\tilde{x}^{t(i)}$ comes with an associated weight $\tilde{w}^{t(i)}$, and the pair $(\tilde{w}^{t(i)}, \tilde{x}^{t(i)})$ is called a particle. In the forecast step a forecast particle $(w^{t(i)}, x^{t(i)})$ is generated from each filtering particle $(\tilde{w}^{t-1(i)}, \tilde{x}^{t-1(i)})$ by generating $x^{t(i)}$ from $\tilde{x}^{t-1(i)}$ as discussed above and by keeping the weight unchanged, that is, $w^{t(i)} = \tilde{w}^{t-1(i)}$. The updating step consists of two parts. First the weights are updated by multiplying each forecast weight $w^{t(i)}$ by the associated likelihood value $p(y^t|x^{t(i)})$, keeping the x^t component of the particles unchanged. Thereafter a resampling may be performed, where $(\tilde{w}^{t(i)}, \tilde{x}^{t(i)})$, $i = 1, \dots , M$ are generated by sampling the $\tilde{x}^{t(i)}$'s independently from $x^{t(i)}$, $i = 1, \dots , M$ with probabilities proportional to the updated weights, and thereafter setting all the new filtering weights $\tilde{w}^{t(i)}$ equal to one. Different criteria can be used to decide whether or not the resampling should be done. The particle filter is very general in that it can be formulated for any Markov x^t process and any observation distribution $p(y^t|x^t)$. However, when running the particle filter one quite often ends up with particle depletion, meaning that a significant fraction of the particles ends up with negligible weights, which in practice requires the number of particles to grow exponentially with the dimension of the state vector x^t. To cope with the particle depletion problem various modifications of the basic particle filter described here have been proposed, for example, the equivalent-weights particle filter of van Leeuwen (2010, 2011).

The ensemble Kalman filter (Burgers, van Leeuwen, & Evensen, 1998; Evensen, 1994) uses approximations in the update step, and thereby produces only an approximate solution to the filtering problem. In the update step it starts by using the forecast samples $x^{t(i)}, i = 1, \dots , M$, to estimate a Gaussian approximation to the forecast distribution $p(x^t|y^{1:t-1})$. This is combined with an assumed Gaussian observation distribution $p(y^t|x^t)$ to obtain a Gaussian approximation to the filtering distribution $p(x^t|y^{1:t})$. Based on this Gaussian approximation the filtering ensemble is generated by sampling $\tilde{x}^{t(i)}, i = 1, \dots , M$ independently from Gaussian distributions, where the mean of $\tilde{x}^{t(i)}$ equals $x^{t(i)}$ plus a shift which depends on the approximate Gaussian filtering distribution. The associated variance is chosen so that the marginal distribution of the generated filtering sample $\tilde{x}^{t(i)}$ is equal to the Gaussian approximation to $p(x^t|y^{1:t})$ when the forecast sample

$x^{t(i)}$ is assumed to be distributed according to the Gaussian approximation to $p(x^t|y^{1:t-1})$. The basic ensemble Kalman filter described here is known to have a tendency to underestimate the variance in the filtering distribution and various remedies have been proposed to correct for this, see for example the discussions in Anderson (2007a, 2007b) and Sætrom and Omre (2013). The square root filter (Tippett, Anderson, Bishop, & Hamill, 2003; Whitaker & Hamill, 2002) is a special variant of the ensemble Kalman filter where the update step is deterministic. The filtering ensemble is then generated from the forecast ensemble only by adding a shift to each ensemble element. Here the size of the shift is chosen so that the marginal distribution of the filtering realizations is equal to the approximated Gaussian filtering distribution.

The Gaussian approximations used in the ensemble Kalman filter limit the use of this filter type to continuous variables, whereas the particle filter setup can be used for both continuous and categorical variables. In the literature there exists a few attempts to use the ensemble Kalman filter setup also for categorical variables, see in particular Oliver, Chen, and Nævdal (2011). The strategy then used for the update step is first to map the categorical variables over to continuous variables, perform the update step as before in the continuous space, and finally map the updated continuous variables back to corresponding categorical variables. In the present article, our goal is to study how the basic ensemble Kalman filter idea can be used for categorical variables without having to map the categorical variables over to a continuous space. As discussed above the update step is the difficult one in ensemble filtering methods. The basic ensemble Kalman filter update starts by estimating a Gaussian approximation to the forecast distribution $p(x^t|y^{1:t-1})$. More generally one may use another parametric class than the Gaussian. For categorical variables the simplest alternative is to consider a first-order Markov chain, which is what we focus on in this article. Having a computationally feasible approximation for the forecast distribution we can find a corresponding approximate filtering distribution. Given the forecast ensemble the question then is from which distribution to simulate the filtering ensemble to obtain that the filtering realizations marginally are distributed according to the given approximate filtering distribution, corresponding to the property for the standard ensemble Kalman filter. In this article we develop in detail an approximate way to do this when the elements of the state vector are binary variables, the approximate forecast distribution is a first-order Markov chain, and the observation distribution has a specifically simple form.

The article has the following layout. First, in Section 2, we review the general state-space model, the associated filtering problem, and present the ensemble Kalman filter. Next, in Section 3, we describe a general ensemble updating framework. Then, in Section 4, we restrict the focus to a situation where the elements of the state vector are binary variables and develop in detail an algorithm for how to perform the update step in this case. After that, we present two numerical experiments with simulated data in Section 5. Finally, in Section 6, we give a few closing remarks and briefly discuss how the proposed updating method for binary vectors can be generalized to a situation with more than two classes and an assumed higher order Markov chain model for the forecast distribution.

2 | PRELIMINARIES

In this section, we review some basic theoretical aspects of ensemble-based filtering methods. The material presented should provide the reader with the necessary background for understanding the proposed approach and it also establishes some of the notations used throughout the article.

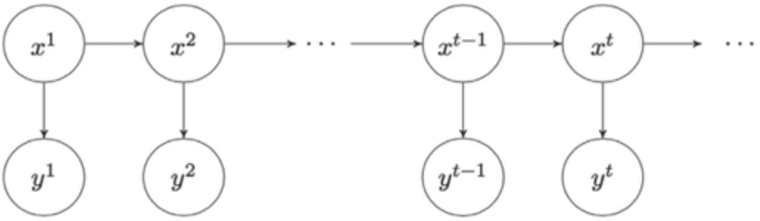


FIGURE 1 Graphical illustration of the state-space model behind the filtering problem

2.1 | Review of the filtering problem

The filtering problem in statistics can be nicely illustrated with a graphical model, see Figure 1. Here, $\{x^t\}_{t=1}^{\infty}$ represents a time series of unobserved states and $\{y^t\}_{t=1}^{\infty}$ a corresponding time series of observations. Each state x^t is n-dimensional and can take on values in a state space Ω_X, while each observation y^t is k-dimensional and can take on values in a state space Ω_Y. The series of unobserved states, called the state process, constitutes a first-order Markov chain with initial distribution $p(x^1)$ and transition probabilities $p(x^t|x^{t-1}), t > 1$. For each state $x^t, t \geq 1$, there is a corresponding observation y^t. The observations are assumed conditionally independent given the state process, with y^t depending on $\{x^t\}_{t=1}^{\infty}$ only through x^t, according to some likelihood model $p(y^t|x^t)$. To summarize, the model is specified by

$$\begin{aligned}
&x^1 \sim p(x^1),\\
&x^t|x^{t-1} \sim p(x^t|x^{t-1}), \quad t > 1,\\
&y^t|x^t \sim p(y^t|x^t), \quad t \geq 1.
\end{aligned}$$

The objective of the filtering problem is, for each t, to compute the so-called filtering distribution, $p(x^t|y^{1:t})$, that is, the distribution of x^t given all observations up to this time, $y^{1:t} = (y^1, \dots, y^t)$. Because of the particular assumptions about the state and observation processes, it can be shown (see Künsch, 2000) that the series of filtering distributions can be computed recursively according to the following equations:

$$\text{i)} \quad p(x^t|y^{1:t-1}) = \int_{\Omega_X} p(x^t|x^{t-1})p(x^{t-1}|y^{1:t-1})\mathrm{d}x^{t-1}, \tag{1a}$$

$$\text{ii)} \quad p(x^t|y^{1:t}) = \frac{p(x^t|y^{1:t-1})p(y^t|x^t)}{\int_{\Omega_X} p(x^t|y^{1:t-1})p(y^t|x^t)\mathrm{d}x^t}. \tag{1b}$$

As one can see, the recursions evolve as a two-step process, each iteration consisting of (i) a prediction step and (ii) an update step. In the prediction, or forecast step, one computes the predictive, or forecast, distribution $p(x^t|y^{1:t-1})$, while in the update step, one computes the filtering distribution $p(x^t|y^{1:t})$ by conditioning the predictive distribution on the incoming observation y^t through application of Bayes' rule. The update step can be formulated as a standard Bayesian inference problem, with $p(x^t|y^{1:t-1})$ becoming the prior, $p(y^t|x^t)$ the likelihood, and $p(x^t|y^{1:t})$ the posterior.

There are two important special cases where the analytical solutions to the filtering recursions in (1a) and (1b) can be computed exactly. The first case is the hidden Markov model (HMM). Here, the state space Ω_X consists of a finite number of states, and the integrals in (1a) and (1b) reduce to finite sums. If the number of states in Ω_X is large; however, the summations become computer-intensive, rendering the filtering recursions *computationally* intractable. The second case is the linear-Gaussian state space model, which can be formulated as follows:

$$
\begin{aligned}
x^1 &\sim \mathcal{N}_n(x^1|\mu^1, \Sigma^1), \\
x^t|x^{t-1} &= A^t x^{t-1} + \omega^t, \quad \omega^t \sim \mathcal{N}_n(\omega|0, \Sigma^t), \\
y^t|x^t &= H^t x^t + \epsilon^t, \quad \epsilon^t \sim \mathcal{N}_k(\epsilon|0, R^t),
\end{aligned} \tag{2}
$$

where $A^t \in \mathbb{R}^{n\times n}$ and $H^t \in \mathbb{R}^{k\times n}$ are nonrandom linear operators, $\Sigma^t \in \mathbb{R}^{n\times n}$ and $R^t \in \mathbb{R}^{k\times k}$ are covariance matrices, and $x^1, \epsilon^1, \epsilon^2, \dots, \omega^1, \omega^2, \dots$ are all independent. In this case, the predictive and filtering distributions are all Gaussian, and the filtering recursions lead to the famous Kalman filter (Kalman, 1960).

In general, we are unable to evaluate the integrals in (1a) and (1b). Approximate solutions therefore become necessary. The most common approach in this regard is the class of ensemble-based methods where a set of samples, called an ensemble, is used to empirically represent the sequence of forecast and filtering distributions. Starting from an initial ensemble $\{x^{1(1)}, \dots, x^{1(M)}\}$ of M independent realizations from the Markov chain initial model $p(x^1)$, the idea is to advance this ensemble forward in time according to the model dynamics. As the original filtering recursions, the propagation of the ensemble alternate between an update step and a prediction step. Specifically, suppose at time $t \geq 1$ that an ensemble $\{x^{t(1)}, \dots, x^{t(M)}\}$ of independent realizations from the forecast distribution $p(x^t|y^{1:t-1})$ is available. We then want to update this forecast ensemble by conditioning on the incoming observation y^t in order to obtain an updated, or posterior, ensemble $\{\tilde{x}^{t(1)}, \dots, \tilde{x}^{t(M)}\}$ with independent realizations from the filtering distribution $p(x^t|y^{1:t})$. If we are able to carry out this updating, we can proceed and propagate the updated ensemble $\{\tilde{x}^{t(1)}, \dots, \tilde{x}^{t(M)}\}$ one time step forward by simulating $x^{t+1(i)}|\tilde{x}^{t(i)} \sim p(x^{t+1}|\tilde{x}^{t(i)})$ for each i. This produces a new forecast ensemble, $\{x^{t+1(1)}, \dots, x^{t+1(M)}\}$, with independent realizations from the forecast distribution $p(x^{t+1}|y^{1:t})$. However, while we are typically able to cope with the forecast step, there is no straightforward way for carrying out the update of the prior ensemble $\{x^{t(1)}, \dots, x^{t(M)}\}$ to a posterior ensemble $\{\tilde{x}^{t(1)}, \dots, \tilde{x}^{t(M)}\}$. Therefore, ensemble methods require approximations in the update step. Consequently, the assumption we make at the beginning of each time step t, that is, that $x^{t(1)}, \dots, x^{t(M)}$ are exact and independent realizations from $p(x^t|y^{1:t-1})$, holds only approximately, except in the initial time step.

In the remains of this article, we focus primarily on the challenging updating of a prior ensemble $\{x^{t(1)}, \dots, x^{t(M)}\}$ to a posterior ensemble $\{\tilde{x}^{t(1)}, \dots, \tilde{x}^{t(M)}\}$ at a specific time step t. We refer to this task as the ensemble updating problem. For simplicity, we omit from now on the time superscript t and the $y^{1:t-1}$ from the notations as these quantities remain fixed. That is, we write x instead of x^t, $p(x)$ instead of $p(x^t|y^{1:t-1})$, $p(x|y)$ instead of $p(x^t|y^{1:t})$, and so on.

2.2 | The ensemble Kalman filter

The ensemble Kalman filter (EnKF), first introduced in the geophysics literature by Evensen (1994), is an approximate ensemble-based method that relies on Gaussian approximations to overcome the difficult updating of the prior ensemble. The updating is done in terms of a linear shift of each ensemble member, closely related to the traditional Kalman filter update.

The literature on the EnKF is extensive, but some basic references include Burgers et al. (1998) and Evensen (2009). Here, we only provide a brief presentation. For simplicity, we restrict the focus to the linear-Gaussian observational model in (2) which, if we omit the superscript t, can be rewritten

$$y|x = Hx + \epsilon, \quad \epsilon \sim \mathcal{N}_k(\epsilon; 0, R).$$

There exist two main classes of EnKFs, stochastic filters and deterministic, or so-called square root filters, differing in whether the updating of the ensemble is carried out stochastically or deterministically. The stochastic EnKF is the most common version, and we begin our below presentation of the EnKF by focusing on this method.

Consider first a linear-Gaussian state space model as introduced in the previous section. Under this linear-Gaussian model, it follows from the Kalman filter recursions that the current forecast, or prior, model $p(x)$ is a Gaussian distribution, $\mathcal{N}_n(x; \mu, \Sigma)$, with analytically tractable mean μ and analytically tractable covariance Σ. Furthermore, the current filtering, or posterior model $p(x|y)$ is a Gaussian distribution, $\mathcal{N}_n(x; \tilde{\mu}, \tilde{\Sigma})$, with mean $\tilde{\mu}$ and covariance $\tilde{\Sigma}$ analytically available from the Kalman filter update equations as

$$\tilde{\mu} = \mu + K(y - H\mu)$$

and

$$\tilde{\Sigma} = (I - KH)\Sigma,$$

respectively, where $K = \Sigma H'(H\Sigma H' + R)^{-1}$ is the Kalman gain. The stochastic EnKF update is based on the following fact: If $x \sim \mathcal{N}_n(x; \mu, \Sigma)$ and $\epsilon \sim \mathcal{N}_k(\epsilon; 0, R)$ are independent random samples, then

$$\tilde{x} = x + K(y - Hx + \epsilon) \tag{3}$$

is a random sample from $\mathcal{N}_n(x; \tilde{\mu}, \tilde{\Sigma})$. The verification of this result is straightforward. Clearly, under the assumption that the prior ensemble $\{x^{(1)}, \dots, x^{(M)}\}$ contains independent samples from the Gaussian distribution $\mathcal{N}_n(x; \mu, \Sigma)$, one theoretically valid way to obtain the updated ensemble is to simulate $\epsilon^{(i)} \sim \mathcal{N}_k(\epsilon; 0, R)$ and replace (x, ϵ) in (3) by $(x^{(i)}, \epsilon^{(i)})$. The stochastic EnKF performs an approximation to this update. Specifically, each prior sample $x^{(i)}$ is updated with a linear shift identical to (3), but with the true Kalman gain K replaced with an empirical estimate $\hat{K}$ inferred from the prior ensemble,

$$\tilde{x}^{(i)} = x^{(i)} + \hat{K}(y - Hx^{(i)} + \epsilon^{(i)}), \quad i = 1, \dots, M. \tag{4}$$

In the EnKF literature, each term $Hx^{(i)} - \epsilon^{(i)}$ is typically referred to as a perturbed observation. Under the linear-Gaussian assumptions, the update in (4) returns approximate samples from the Gaussian posterior model $\mathcal{N}_n(x; \tilde{\mu}, \tilde{\Sigma})$. The update is in this case consistent in the sense that as the ensemble size goes to infinity, the distribution of the updated samples converges to $\mathcal{N}_n(x; \tilde{\mu}, \tilde{\Sigma})$, that is, the solution of the Kalman filter.

Although the EnKF update is based on linear-Gaussian assumptions about the underlying model, it can still be applied in nonlinear, non-Gaussian situations. Naturally, bias is in this case introduced, and the updated samples will not converge in distribution to the true posterior $p(x|y)$. However, since the update is a linear combination of the $x^{(i)}$'s, non-Gaussian properties present in the true prior and posterior models can, to some extent, be captured.

Deterministic EnKFs instead use a nonrandom linear transformation to update the ensemble. In the following, let $\hat{\mu}$ and $\hat{\Sigma}$ denote estimates of μ and Σ, respectively, obtained from the prior ensemble. Furthermore, let $\hat{\hat{\mu}}$ and $\hat{\hat{\Sigma}}$ denote the mean and covariance, respectively, of the Gaussian posterior model $\mathcal{N}_n(x; \hat{\hat{\mu}}, \hat{\hat{\Sigma}})$ corresponding to the Gaussian prior approximation $\mathcal{N}_n(x; \hat{\mu}, \hat{\Sigma})$. Generally, the update equation of a square root EnKF can be written as

$$\tilde{x}^{(i)} = \hat{\mu} + \hat{K}(y - H\hat{\mu}) + B(x^{(i)} - \hat{\mu}), \quad i = 1, \dots, M, \tag{5}$$

where $B \in \mathbb{R}^{n\times n}$ is a solution to the quadratic matrix equation

$$B\hat{\Sigma}B' = (I - \hat{K}H)\hat{\Sigma}.$$

Note that B is not unique except in the univariate case. This gives rise to a variety of square root algorithms, see Tippett et al. (2003). As such, several square root formulations have been proposed in the literature, including, but not limited to, Anderson (2001), Bishop, Etherton, and Majumdar (2001), and Whitaker and Hamill (2002). The nonrandom square root EnKF update in (5) ensures that the sample mean and sample covariance of the posterior ensemble equal $\hat{\hat{\mu}}$ and $\hat{\hat{\Sigma}}$ *exactly*. This is different from stochastic EnKFs where, under linear-Gaussian assumptions, the sample mean and sample covariance of the posterior ensemble only equal $\hat{\hat{\mu}}$ and $\hat{\hat{\Sigma}}$ in expectation.

3 | A GENERAL ENSEMBLE UPDATING FRAMEWORK

In this section, we present a general ensemble updating framework. Both the EnKF and the updating procedure for binary vectors proposed in this article can be viewed as special applications of the framework.

3.1 | The framework

For convenience, we first give a brief review of the ensemble updating problem. Starting out, we have a prior ensemble, $\{x^{(1)}, \dots, x^{(M)}\}$, which is assumed to contain independent realizations from a prior model $p(x)$. The prior model $p(x)$ is typically intractable in this context, either computationally or analytically, or both. Given an observation y and a corresponding likelihood model $p(y|x)$ the goal is to update the prior ensemble according to Bayes' rule in order to obtain a posterior ensemble, $\{\tilde{x}^{(1)}, \dots, \tilde{x}^{(M)}\}$, with independent realizations from the posterior model $p(x|y)$. However, carrying out this update exactly is generally unfeasible and approximate strategies are required.

Conceptually, the proposed framework is quite simple. It involves three main steps as follows. First, we replace the intractable model $p(x|y) \propto p(x)p(y|x)$ with a simpler model $f(x|y) \propto f(x)p(y|x)$. Here, $f(x)$ is an approximation to the prior $p(x)$ and is constructed from the samples of the prior ensemble, while $f(x|y)$ is the corresponding posterior distribution which follows from Bayes' rule. In the remains of this article, we refer to the model $f(x|y) \propto f(x)p(y|x)$ as the *assumed* model. Notice that the likelihood model $p(y|x)$ has not been replaced; for simplicity, we assume that this model already has a convenient form. Second, we put forward a distribution conditional on x and y, denoted $q(\tilde{x}|x,y)$, obeying the following property:

$$f(\tilde{x}|y) = \int_{\Omega_X} f(x)q(\tilde{x}|x,y)\mathrm{d}x. \tag{6}$$

Third, we update the prior ensemble by generating samples from this conditional distribution,

$$\tilde{x}^{(i)} \sim q(\tilde{x}|x^{(i)},y), \quad i = 1, \dots ,M.$$

To understand the framework, note that under the assumption that the assumed model is correct, the prior samples have distribution $f(x)$ and the updated samples should have distribution $f(x|y)$. If one is able to compute and sample from $f(x|y)$, one straightforward way to obtain the updated samples is to sample directly from $f(x|y)$. However, since the assumed model is not really the correct one, this is probably not the best way to proceed. The prior ensemble contains valuable information about the true model $p(x)$ that may not have been captured by the assumed model $f(x)$, and by straightforward simulation from $f(x|y)$ this information is lost. To capture more information from the prior ensemble, it is advantageous to simulate conditionally on the prior samples. This is why we introduce the conditional distribution $q(\tilde{x}|x,y)$. The criterion in (6) ensures that the marginal distribution of each updated sample $\tilde{x}^{(i)}$ generated by $q(\tilde{x}|x,y)$ still is $f(x|y)$ given that the assumed model is correct. However, since the assumed model is not the correct model, the marginal distribution of the updated samples is not $f(x|y)$, but some other distribution, hopefully one closer to the true posterior model $p(x|y)$.

There are two especially important things about the proposed framework that must be taken care of in a practical application. First, we need to select an assumed prior $f(x)$ which, combined with the likelihood model $p(y|x)$, returns a tractable posterior $f(x|y)$. Second, we need to construct the updating distribution $q(\tilde{x}|x,y)$. Typically, there are many, or infinitely many, distributions $q(\tilde{x}|x,y)$ which all fulfill the constraint in (6). A natural strategy for choosing a solution $q(\tilde{x}|x,y)$ is then to define a criterion of optimality and set $q(\tilde{x}|x,y)$ equal to the corresponding optimal solution. Below, we present two special cases of the proposed framework. The first case corresponds to the EnKF where $f(x)$, $p(y|x)$, and $q(\tilde{x}|x,y)$ are all Gaussian distributions. In the second case, $f(x)$ and $p(y|x)$ constitute a hidden Markov model with binary states $x_i \in \{0,1\}$, and the updating distribution $q(\tilde{x}|x,y)$ is a transition matrix.

3.2 | The EnKF as a special case

The EnKF can be seen as a special case of the proposed framework. The assumed prior model $f(x)$ is in this case a Gaussian distribution. Combined with a linear-Gaussian likelihood model $p(y|x)$ the corresponding assumed posterior model $f(x|y)$ is also Gaussian. The conditional distribution $q(\tilde{x}|x,y)$ in the EnKF arises from the linear update, and takes a different form depending on whether the filter is stochastic or deterministic. In stochastic EnKF, the linear update (4) yields a Gaussian distribution $q(\tilde{x}|x,y)$ with mean equal to $x + \hat{K}(y - Hx)$ and covariance equal to $\hat{K}R\hat{K}'$, that is,

$$q(\tilde{x}|x,y) = \mathcal{N}(\tilde{x}; x + \hat{K}(y - Hx), \hat{K}R\hat{K}').$$

In square root EnKF, the case is a bit different. Because the linear update in (5) is deterministic, $q(\tilde{x}|x,y)$ has zero covariance and becomes a degenerate Gaussian distribution, or a delta function, located at the value to which x is moved, that is

$$q(\tilde{x}|x,y) = \delta(\tilde{x}; \hat{\mu} + \hat{K}(y - H\hat{\mu}) + B(x \ - \hat{\mu})).$$

As mentioned in Section 2.2, the matrix B in square root EnKF is not unique except in the univariate case. This gives rise to a class of square root EnKF algorithms. When choosing a particular filter, one could proceed as briefly suggested at the end of Section 3.1 and choose the matrix B so that it is optimal with respect to some criterion.

3.3 | The proposed method for binary vectors as a special case

Suppose $x = (x_1, \dots, x_n)$ is a vector of n binary variables, $x_i \in \{0,1\}$, and that x is spatially arranged along a line. A possible assumed prior model for x is then a first-order Markov chain,

$$f(x) = f(x_1)f(x_2|x_1)\cdots f(x_n|x_{n-1}).$$

Furthermore, suppose that for each variable x_i there is a corresponding observation, y_i, so that $y = (y_1, \dots, y_n)$, and suppose that the y_i's are conditionally independent given x, with y_i depending on x only through x_i,

$$p(y|x) = p(y_1|x_1)\cdots p(y_n|x_n).$$

This combination of $f(x)$ and $p(y|x)$ constitutes a hidden Markov model as introduced in Section 2. It follows that the corresponding assumed posterior model $f(x|y)$ is also a first-order Markov chain for which all quantities of interest are possible to compute. Note that we can also handle likelihood models $p(y|x)$ where only a selection of the x_i's are observed, as long as the observed y_j's are conditionally independent and each y_j is only connected to one variable x_i of x.

Now, since $\Omega_X = \{0,1\}^n$ is a discrete sample space, we rewrite the constraint in (6) as a sum,

$$f(\tilde{x}|y) = \sum_{x\in\Omega_X} f(x)q(\tilde{x}|x,y). \tag{7}$$

Because of the discrete context, $q(\tilde{x}|x,y)$ represents a transition matrix, not a density as in EnKF. The size of this transition matrix is $2^n \times 2^n$ since there are 2^n possible configurations of the state vector x. Brute force, the specification of $q(\tilde{x}|x,y)$ involves the specification of $2^n(2^n - 1)$ parameters, and the constraint in (7) leads to a system of $2^n - 1$ linear equations in these parameters. The number of unknowns (parameters) is larger than the number of equations, so there are infinitely many valid solutions of $q(\tilde{x}|x,y)$. To choose a specific solution, we proceed as suggested in Section 3.1 and seek a solution which is optimal with respect to a certain criterion; we consider this in full detail in the next section.

Even for moderate n, dealing with the problem outlined above is too complicated. Therefore, we need to settle with an approximate approach. Specifically, instead of seeking a solution $q(\tilde{x}|x,y)$ which retains the whole Markov chain model $f(x|y)$ cf. the constraint (7), we pursue a solution which only retains all the marginal distributions $f(x_i,x_{i+1}|y)$ of $f(x|y)$. For convenience, let

$$\pi(\tilde{x}, x|y) = f(x)q(\tilde{x}|x, y) \tag{8}$$

denote the distribution of x and $\tilde{x}$ under the assumption that x is distributed according to $f(x)$ and $\tilde{x}$ is generated from $q(\tilde{x}|x, y)$. Mathematically, the requirement that $q(\tilde{x}|x, y)$ must retain all the marginal distributions $f(x_i,x_{i+1}|y)$ can then be expressed as

$$\pi(\tilde{x}_i, \tilde{x}_{i+1}|y) = f(\tilde{x}_i, \tilde{x}_{i+1}|y), \quad i = 1, \dots, n-1. \tag{9}$$

In the next section, we consider in full detail how to compute a distribution $q(\tilde{x}|x, y)$ which fulfills (9). In particular, we impose Markov properties on $q(\tilde{x}|x, y)$, formulate an optimality criterion for $q(\tilde{x}|x, y)$, and use dynamic programming to construct the optimal solution.

4 | ENSEMBLE UPDATING OF BINARY STATE VECTORS

This section continues on the situation introduced in Section 3.3. The main focus is on the construction of the updating distribution $q(\tilde{x}|x, y)$. In Section 4.1 we formulate an optimality criterion and enforce Markov properties on $q(\tilde{x}|x, y)$. Thereafter, in Section 4.2, we present a dynamic programming (DP) algorithm for constructing the optimal solution of $q(\tilde{x}|x, y)$. Finally, in Section 4.3, we take a closer look at some more technical aspects of the DP algorithm.

4.1 | Optimality criterion

As mentioned in the previous section, there are infinitely many valid solutions of $q(\tilde{x}|x, y)$. For us, however, it is sufficient with *one* solution, preferably an *optimal* solution, $q^*(\tilde{x}|x, y)$, with respect to some criterion. To specify an appropriate optimality criterion, we argue that in order for $q(\tilde{x}|x, y)$ to retain information from the prior ensemble and capture important properties of the true prior and posterior models, it should not make unnecessary changes to the prior samples. That is, as we update each prior sample $x^{(i)}$, we should take new information from the incoming observation y into account and, to a certain extent, push $x^{(i)}$ toward y, but the adjustment we make should be minimal. We therefore propose to define the optimal solution $q^*(\tilde{x}|x, y)$ as the one that maximizes the expected number of variables, or components, of x that remain unchanged after the update to $\tilde{x}$. Mathematically, that is

$$q^*(\tilde{x}|x, y) = \underset{q(\tilde{x}|x,y)}{\text{argmax}}\ \mathrm{E}_\pi \left[\sum_{i=1}^{n} 1(x_i = \tilde{x}_i) \right], \tag{10}$$

where the subscript π is used to indicate that the expectation is taken over the joint distribution $\pi(\tilde{x}, x|y)$ in (8).

The problem of computing the optimal solution $q^*(\tilde{x}|x, y)$ in (10) given the original constraint in (7) can be interpreted as a discrete version of an optimal transport problem (Villani, 2009). Brute force, the optimization problem is a linear programming problem since (10) defines an objective function which is linear in $q(\tilde{x}|x, y)$ and (7) yields a set of equations that are linear in $q(\tilde{x}|x, y)$. However, since the number of variables involved is so large, the problem is too demanding to cope

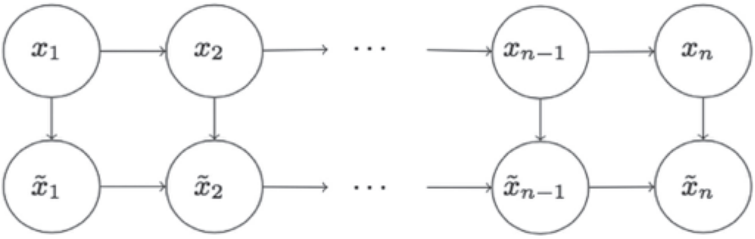


FIGURE 2 Graphical illustration of the updating distribution $q(\tilde{x}|x, y)$

with. Therefore, we resort to an approximate approach. As mentioned in the previous section, we replace the requirement (7) with the less strict requirement (9). Moreover, to reduce the number of parameters involved, we enforce Markov properties on $\pi(\tilde{x}, x|y)$ as illustrated graphically in Figure 2. Given this structure, $q(\tilde{x}|x, y)$ can be factorized as

$$q(\tilde{x}|x, y) = q(\tilde{x}_1|x_1, y)q(\tilde{x}_2|\tilde{x}_1, x_2, y)q(\tilde{x}_3|\tilde{x}_2, x_3, y) \cdots q(\tilde{x}_n|\tilde{x}_{n-1}, x_n, y). \tag{11}$$

Consequently, the number of parameters reduces from $2^n(2^n - 1) = \mathcal{O}(4^n)$ to $2 + 4(n - 1) = \mathcal{O}(n)$, namely, two parameters for the first factor $q(\tilde{x}_1|x_1, y)$, and four parameters for each $q(\tilde{x}_k|\tilde{x}_{k-1}, x_k, y)$, $k = 2, \ldots, n$. Another, and just as important, consequence of the Markov properties is that the optimal solution $q^*(\tilde{x}|x, y)$ can be efficiently computed using dynamic programming. Following (11), the optimal solution can be factorized as

$$q^*(\tilde{x}|x, y) = q^*(\tilde{x}_1|x_1, y)q^*(\tilde{x}_2|\tilde{x}_1, x_2, y)q^*(\tilde{x}_3|\tilde{x}_2, x_3, y) \cdots q^*(\tilde{x}_n|\tilde{x}_{n-1}, x_n, y). \tag{12}$$

The next section presents a DP algorithm where the n factors in (12) are constructed recursively.

4.2 | Dynamic programming

Here, we describe a DP algorithm for constructing the optimal solution $q^*(\tilde{x}|x, y)$ introduced in the previous section. The algorithm involves a backward recursion and a forward recursion. The main challenge is the backward recursion and the details therein are a bit technical. For simplicity, this section provides an overall description of the algorithm, while the more technical aspects of the backward recursion are considered separately in Section 4.3. Following the notation introduced in (8), we use the notation $\pi(\tilde{x}_{i:j}, x_{k:l}|y)$, $1 \leq i \leq j \leq n$, $1 \leq k \leq l \leq n$, to denote the joint distribution of $\tilde{x}_{i:j} = (\tilde{x}_i, \ldots, \tilde{x}_j)$ and $x_{k:l} = (x_k, \ldots, x_l)$ under the assumption that x is distributed according to $f(x)$ and $\tilde{x}$ is simulated using $q(\tilde{x}|x, y)$. Furthermore, we introduce the following simplifying notations:

$$\pi_k = \begin{cases} \pi(x_1|y), & k = 1, \\ \pi(\tilde{x}_{k-1}, x_k|y), & 2 \leq k \leq n, \end{cases}$$

$$q_k = \begin{cases} q(\tilde{x}_1|x_1, y), & k = 1, \\ q(\tilde{x}_k|\tilde{x}_{k-1}, x_k, y), & 2 \leq k \leq n. \end{cases}$$

The backward recursion of the DP algorithm involves recursive computation of the quantities

$$\max_{q_{k:n}} \mathrm{E}_\pi \left[\sum_{i=k}^{n} 1(x_i = \tilde{x}_i) \right] \tag{13}$$

for $k = n, n-1, \dots, 1$. In words, (13) represents the largest possible contribution of the partial expectation $\mathrm{E}_\pi \left[\sum_{i=k}^{n} 1(x_i = \tilde{x}_i) \right]$ to the full expectation $\mathrm{E}_\pi \left[\sum_{i=1}^{n} 1(x_i = \tilde{x}_i) \right]$ that can be obtained for a fixed $\pi(\tilde{x}_{1:k-1}, x_{1:k}|y)$. The recursion uses the fact that, for $k \geq 2$, the Markov properties of $\pi(\tilde{x}, x|y)$ yield

$$\begin{aligned} \max_{q_{(k-1):n}} \mathrm{E}_\pi \left[\sum_{i=k-1}^{n} 1(x_i = \tilde{x}_i) \right] &= \max_{q_{(k-1):n}} \mathrm{E}_\pi \left[1(x_{k-1} = \tilde{x}_{k-1}) + \sum_{i=k}^{n} 1(x_i = \tilde{x}_i) \right] \\ &= \max_{q_{k-1}} \left[\mathrm{E}_\pi [1(x_{k-1} = \tilde{x}_{k-1})] + \max_{q_{k:n}} \mathrm{E}_\pi \left[\sum_{i=k}^{n} 1(x_i = \tilde{x}_i) \right] \right] \end{aligned} \tag{14}$$

suggesting that the full maximum value in (10) can be computed recursively by recursive maximization over $q_n, q_{n-1}, \dots, q_1$.

An essential aspect of the backward recursion are the distributions $\pi_1, \dots, \pi_n$. At each step k, we compute (13) as a function of π_k. Essentially, each π_k, $k \geq 2$, consists of four numbers, or parameters, one for each possible configuration of the pair $(\tilde{x}_{k-1}, x_k)$. However, one parameter is lost since $\pi(\tilde{x}_{k-1}, x_k|y)$ is a distribution so that the four numbers must sum to one. Another two parameters are lost since we require that $\pi(\tilde{x}_{k-1}, x_k|y)$ retains the marginal distributions $f(\tilde{x}_{k-1}|y)$ and $f(x_k)$, that is, we require

$$\sum_{\tilde{x}_{k-1}} \pi(\tilde{x}_{k-1}, x_k|y) = f(x_k)$$

and

$$\sum_{x_k} \pi(\tilde{x}_{k-1}, x_k|y) = f(\tilde{x}_{k-1}|y).$$

Thereby only one parameter, which in the following we denote by t_k, remains. This parameter t_k is free to vary within an interval $[t_k^{\min}, t_k^{\max}]$, where the bounds $t_k^{\min}$ and $t_k^{\max}$ are determined by the probabilistic nature of π_k. An example parametrization is to set $t_k = \pi(\tilde{x}_{k-1} = 0, x_k = 0|y)$, which is the approach taken in this work. Below, the notation $\pi_{t_k}(\tilde{x}_{k-1}, x_k|y)$ will, when appropriate, be used instead of $\pi(\tilde{x}_{k-1}, x_k|y)$, in order to express the dependence on t_k more explicitly. The chosen parameter t_k leads to a parametrization of π_k as follows,

$$\begin{aligned} &\pi_{t_k}(\tilde{x}_{k-1} = 0, x_k = 0|y) = t_k, \\ &\pi_{t_k}(\tilde{x}_{k-1} = 0, x_k = 1|y) = f(\tilde{x}_{k-1} = 0|y) - t_k, \\ &\pi_{t_k}(\tilde{x}_{k-1} = 1, x_k = 0|y) = f(x_k = 0) - t_k, \\ &\pi_{t_k}(\tilde{x}_{k-1} = 1, x_k = 1|y) = 1 - f(x_k = 0) - f(\tilde{x}_{k-1} = 0|y) + t_k, \end{aligned}$$

and the bounds of the interval $[t_k^{\min}, t_k^{\max}]$ are given as

$$t_k^{\min} = \max\{0, f(x_k = 0) + f(x_{k-1} = 0|y) - 1\}, \tag{15}$$

$$t_k^{\max} = \min\{f(x_k = 0), f(x_{k-1} = 0|y)\}. \tag{16}$$

For $k = 1$, the situation is a bit different, since there is only one variable, x_1, involved in $\pi_1 = \pi(x_1|y)$. In fact, due to (8), we have $\pi(x_1|y) = f(x_1)$. Consequently, t_1 is not a parameter free to vary within a certain range, but a fixed number. Here, we set $t_1 = f(x_1 = 0)$.

Apart from the parametrization of π_k, an essential feature of each π_k, for $k \geq 2$, is its dependence on π_{k-1} and q_{k-1}. This connection is due to the particular structure of $\pi(\tilde{x}, x|y)$. Generally, for $k \geq 3$, we know that π_k, or $\pi(\tilde{x}_{k-1}, x_k|y)$, can be computed by summing out the variables $\tilde{x}_{k-2}$ and x_{k-1} from the joint distribution $\pi(\tilde{x}_{k-2}, \tilde{x}_{k-1}, x_{k-1}, x_k|y)$,

$$\pi(\tilde{x}_{k-1}, x_k|y) = \sum_{\tilde{x}_{k-2}} \sum_{x_{k-1}} \pi(\tilde{x}_{k-2}, \tilde{x}_{k-1}, x_{k-1}, x_k|y), \tag{17}$$

and the distribution $\pi(\tilde{x}_{k-2}, \tilde{x}_{k-1}, x_{k-1}, x_k|y)$ can be written in the particular form

$$\pi(\tilde{x}_{k-2}, \tilde{x}_{k-1}, x_{k-1}, x_k|y) = \pi(\tilde{x}_{k-2}, x_{k-1}|y) q(\tilde{x}_{k-1}|\tilde{x}_{k-2}, x_{k-1}, y) f(x_k|x_{k-1}).$$

Similarly, for the special case $k = 2$, we can compute $\pi(\tilde{x}_1, x_2|y)$ by summing out x_2 from $\pi(\tilde{x}_1, x_1, x_2|y)$,

$$\pi(\tilde{x}_1, x_2|y) = \sum_{x_1} \pi(\tilde{x}_1, x_1, x_2|y), \tag{18}$$

where $\pi(\tilde{x}_1, x_1, x_2|y)$ can be written as

$$\pi(\tilde{x}_1, x_1, x_2|y) = f(x_1) q(\tilde{x}_1|x_1, y) f(x_2|x_1). \tag{19}$$

Inserting $\tilde{x}_{k-1} = 0$ and $x_k = 0$ in (17), and using that π_{k-1} is parametrized by t_{k-1}, we obtain a formula for t_k in terms of t_{k-1} and q_{k-1}, $k \geq 3$. Likewise, inserting $\tilde{x}_1 = 0$ and $x_2 = 0$ in (18), and using that $f(x_1 = 0) = t_1$, we obtain a formula for t_2 in terms of t_1 and q_1. To express the dependence of t_k on t_{k-1} and q_{k-1}, $k \geq 2$, we will use the notation

$$t_k = t_k(t_{k-1}, q_{k-1}).$$

In some of the following equations, it will be necessary to explicitly express that (13) is a function of t_k. We therefore define

$$\mathrm{E}^*_{k:n}(t_k) = \max_{q_{k:n}} \mathrm{E}_\pi \left[\sum_{i=k}^{n} 1(x_i = \tilde{x}_i) \right].$$

Similarly, we need a notation for the argument of the maximum in (14) as a function of t_k:

$$q^*_{t_k}(\tilde{x}_k|\tilde{x}_{k-1}, x_k, y) = \underset{q_k}{\operatorname{argmax}} \left[\mathrm{E}_\pi[1(x_k = \tilde{x}_k)] + \max_{q_{(k+1):n}} \mathrm{E}_\pi \left[\sum_{i=k}^{n} 1(x_i = \tilde{x}_i) \right] \right], \quad 2 \leq k \leq n,$$

$$q^*_{t_1}(\tilde{x}_1|x_1, y) = \underset{q_1}{\operatorname{argmax}} \left[\mathrm{E}_\pi[1(x_1 = \tilde{x}_1)] + \max_{q_{2:n}} \mathrm{E}_\pi \left[\sum_{i=1}^{n} 1(x_i = \tilde{x}_i) \right] \right].$$

If $q^*_{t_k}(\tilde{x}_k|\tilde{x}_{k-1},x_k,y)$ and $q^*_{t_1}(\tilde{x}_1|x_1,y)$ are discussed in a context where the specific values of the involved variables are not important, simpler notations are preferable. In this regard, we also introduce

$$q^*_k(t_k) = \begin{cases} q^*_{t_k}(\tilde{x}_k|\tilde{x}_{k-1},x_k,y), & 2 \leq k \leq n, \\ q^*_{t_1}(\tilde{x}_1|x_1,y), & k = 1. \end{cases}$$

Also, we need a notation for $\mathrm{E}_\pi[1(x_k = \tilde{x}_k)]$ indicating that this is a function of both t_k and q_k,

$$\mathrm{E}_k(t_k, q_k) = \mathrm{E}_\pi[1(x_k = \tilde{x}_k)].$$

The backward recursion computes $\mathrm{E}^*_{k:n}(t_k)$ recursively for $k = n, n-1, \dots, 1$. Each step performs a maximization over q_k as a function of the parameter t_k. The recursion is initialized by

$$\mathrm{E}^*_n(t_n) = \max_{q_n} [\mathrm{E}_n(t_n, q_n)] \tag{20}$$

and

$$q^*_n(t_n) = \operatorname*{argmax}_{q_n} [\mathrm{E}_n(t_n, q_n)]]. \tag{21}$$

Then, for $k = n-1, n-2, \dots, 1$, the recursion proceeds according to

$$\mathrm{E}^*_{k:n}(t_k) = \max_{q_k} [\mathrm{E}_k(t_k, q_k) + \mathrm{E}^*_{(k+1):n}(t_{k+1}(t_k, q_k))], \tag{22}$$

$$q^*_k(t_k) = \operatorname*{argmax}_{q_k} [\mathrm{E}_k(t_k, q_k) + \mathrm{E}^*_{(k+1):n}(t_{k+1}(t_k, q_k))]. \tag{23}$$

Note that at the final step of the backward recursion, where $k = 1$, we compute $\mathrm{E}^*_{1:n}(t_1)$ and $q^*_1(t_1)$. Now, since we have one specific value for t_1, we also obtain one specific value for $\mathrm{E}^*_{1:n}(t_1)$ and corresponding specific values for $q^*_1(t_1)$. This completes the backward recursion.

After the backward recursion, the forward recursion can proceed. Here, we recursively compute specific values for $t_2, t_3, \dots, t_n$. Hence we recursively obtain the optimal values $q^*(\tilde{x}_2|\tilde{x}_1, x_2, y)$, $q^*(\tilde{x}_3|\tilde{x}_2, x_3, y), \dots, q^*(\tilde{x}_n|\tilde{x}_{n-1}, x_n, y)$ in (12). The forward recursion is initialized by

$$t^*_1 = t_1$$

and

$$q^*(\tilde{x}_1|x_1, y) = q^*_{t^*_1}(\tilde{x}_1|x_1, y).$$

Then, for $k = 2, 3, \dots, n$, the recursion proceeds according to

$$t^*_k = t_k(t^*_{k-1}, q^*_{k-1}(t^*_{k-1})),$$

$$q^*(\tilde{x}_k|\tilde{x}_{k-1}, x_k, y) = q^*_{t^*_k}(\tilde{x}_k|\tilde{x}_{k-1}, x_k, y).$$

When the forward recursion terminates, the optimal solution $q^*(\tilde{x}|x, y)$ is readily available.

4.3 | Parametric, piecewise linear programming

In this section, we look further into the backward recursion of the DP algorithm described in Section 4.2. As we shall see, each step of the recursion involves the setup of an optimization problem that we refer to as a parametric, piecewise linear program, namely, an optimization problem with a piecewise linear objective function subject to a set of linear constraints, which we solve as a function of the parameter t_k. For simplicity of writing, we now introduce the following notations:

$$q_k^{ij} = q(\tilde{x}_k = 0|\tilde{x}_{k-1} = i, x_k = j, y), \tag{24a}$$

$$q_1^{i} = q(\tilde{x}_1 = 0|x_1 = i, y), \tag{24b}$$

$$f_k^{ij} = f(x_{k-1} = i, x_k = j|y), \tag{24c}$$

$$\pi_k^{ij}(t_k) = \pi_{t_k}(\tilde{x}_{k-1} = i, x_k = j|y), \tag{24d}$$

$$q_k^{*ij}(t_k) = q_{t_k}^*(\tilde{x}_k = 0|\tilde{x}_{k-1} = i, x_k = j, y), \tag{24e}$$

$$\rho_{k-1}^{i|j} = f(x_k = i|x_{k-1} = j), \tag{24f}$$

for $i,j \in \{0,1\}$ and $k \geq 2$.

Reconsider the initial step of the backward recursion. The goal of this step is to compute $E_n^*(t_n)$ in (20) and $q_n^*(t_n)$ in (21). The objective function at this step, $\mathrm{E}_n(t_n,q_n)$, can be computed as

$$E_n(t_n, q_n) = \pi_n^{00}(t_n)q_n^{00} + \pi_n^{01}(t_n)(1 - q_n^{01}) + \pi_n^{10}(t_n)q_n^{10} + \pi_n^{11}(t_n)(1 - q_n^{11}). \tag{25}$$

Since $\pi_n^{01}(t_n) + \pi_n^{11}(t_n) = f(x_n = 1)$, we can, after rearranging the terms, rewrite (25) as

$$E_n(t_n, q_n) = \pi_n^{00}(t_n)q_n^{00} - \pi_n^{01}(t_n)q_n^{01} + \pi_n^{10}(t_n)q_n^{10} - \pi_n^{11}(t_n)q_n^{11} + f(x_n = 1). \tag{26}$$

As a function of the parameter $t_n \in [t_n^{\min}, t_n^{\max}]$, we are interested in computing the solution of q_n which maximizes (26). In this regard one needs to take the constraint in (9) into account. Specifically, the constraint entails at this step that

$$\pi(\tilde{x}_{n-1}, \tilde{x}_n|y) = f(\tilde{x}_{n-1}, \tilde{x}_n|y)$$

for all $\tilde{x}_{n-1}, \tilde{x}_n \in \{0, 1\}$. Hence, using that $\pi(\tilde{x}_{n-1}, \tilde{x}_n, x_n|y) = \pi(\tilde{x}_{n-1}, x_n|y)q(\tilde{x}_n|\tilde{x}_{n-1}, x_n, y)$, and that $\pi(\tilde{x}_{n-1}, \tilde{x}_n|y)$ follows by summing out x_n from $\pi(\tilde{x}_{n-1}, \tilde{x}_n, x_n|y)$, we see that q_n must fulfill

$$f(\tilde{x}_{n-1}, \tilde{x}_n|y) = \sum_{x_n} \pi(\tilde{x}_{n-1}, x_n|y)q(\tilde{x}_n|\tilde{x}_{n-1}, x_n, y).$$

This requirement leads to four linear equations of which two are linearly independent, one where we set $\tilde{x}_{n-1} = 0$ and one where we set $\tilde{x}_{n-1} = 1$. Using the notations in (24a)–(24d), the two linearly independent equations can be written as

$$f_n^{00} = \pi_n^{00}(t_n)q_n^{00} + \pi_n^{01}(t_n)q_n^{01}, \tag{27a}$$

$$f_n^{10} = \pi_n^{10}(t_n)q_n^{10} + \pi_n^{11}(t_n)q_n^{11}. \tag{27b}$$

Additionally, we know that $q_n^{00}, q_n^{01}, q_n^{10}$, and q_n^{11} can only take values within the interval [0,1],

$$0 \leq q_n^{ij} \leq 1, \quad \text{for all } \; i,j \in \{0,1\}. \tag{28}$$

To summarize, we want, as a function of the parameter $t_n \in [t_n^{\min}, t_n^{\max}]$, to compute the solutions of $q_n^{00}, q_n^{01}, q_n^{10}$, and q_n^{11} which maximize the function (26) subject to the constraints in (27) and (28). For any fixed t_n, this is a maximization problem where both the objective function and all the constraints are linear in $q_n^{00}, q_n^{01}, q_n^{10}$, and q_n^{11}. As such, the maximization problem can, for a given value of t_n, be formulated as a linear program and solved accordingly. In Appendix A, we show that the optimal solutions $q_n^{*00}(t_n)$, $q_n^{*01}(t_n)$, $q_n^{*10}(t_n)$, and $q_n^{*11}(t_n)$ are piecewise-defined functions of t_n and easy to compute analytically. Furthermore, we show that the corresponding function $E_n^*(t_n)$, obtained by inserting $q_n^{*00}(t_n)$, $q_n^{*01}(t_n)$, $q_n^{*10}(t_n)$, and $q_n^{*11}(t_n)$ into (26), is a continuous piecewise linear (CPL) function of t_n.

Next, consider the intermediate steps of the backward recursion, that is, $k = n-1, n-2, \dots, 2$. At each such step, the aim is to compute $E_{k:n}^*(t_k)$ in (22) and $q_k^*(t_k)$ in (23). The objective function at each step reads

$$E_{k:n}(t_k, q_k) = E_k(t_k, q_k) + E_{(k+1):n}^*(t_{k+1}(t_k, q_k)), \tag{29}$$

and this function is to be maximized with respect to q_k. The first term, $E_k(t_k,q_k)$, in (29) can be computed as

$$E_k(t_k, q_k) = \pi_k^{00}(t_k)q_k^{00} - \pi_k^{01}(t_k)q_k^{01} + \pi_k^{10}(t_k)q_k^{10} - \pi_k^{11}(t_k)q_k^{11} + f(x_k = 1). \tag{30}$$

The second term, $E_{(k+1):n}^*(t_{k+1}(t_k, q_k))$, is a CPL function of t_{k+1}. For $k = n-1$, this result is immediate, since we know from the first iteration that $E_n^*(t_n)$ is CPL. For $k < n-1$, the result is explained in Appendix A. Since $t_{k+1}(t_k,q_k)$ is linear in q_k, it follows that $E_{k+1}^*(t_{k+1}(t_k, q_k))$ is CPL in q_k for any given $t_k \in [t_k^{\min}, t_k^{\max}]$. Hence, the objective function in (29) is also CPL in q_k for any $t_k \in [t_k^{\min}, t_k^{\max}]$. As in the first backward step, we have the following equality and inequality constraints for q_k:

$$f_k^{00} = \pi_k^{00}(t_k)q_k^{00} + \pi_k^{01}(t_k)q_k^{01}, \tag{31a}$$

$$f_k^{10} = \pi_k^{10}(t_k)q_k^{10} + \pi_k^{11}(t_k)q_k^{11} \tag{31b}$$

and

$$0 \leq q_k^{00}, q_k^{01}, q_k^{10}, q_k^{11} \leq 1. \tag{32}$$

Additionally, we now need to incorporate constraints ensuring that q_k and t_k return a value t_{k+1} within the interval $[t_{k+1}^{\min}, t_{k+1}^{\max}]$, where $t_{k+1}^{\min}$ and $t_{k+1}^{\max}$ are given by (15) and (16), respectively.

That is, we require

$$t_{k+1}^{\min} \leq t_{k+1}(t_k, q_k) \leq t_{k+1}^{\max}, \tag{33}$$

where $t_{k+1}(t_k,q_k)$ follows from (17) as

$$t_{k+1}(t_k, q_k) = \pi_k^{00}(t_k)q_k^{00}\rho_k^{0|0} + \pi_k^{01}(t_k)q_k^{01}\rho_k^{0|1} + \pi_k^{10}(t_k)q_k^{10}\rho_k^{0|0} + \pi_k^{11}(t_k)q_k^{11}\rho_k^{0|1}. \tag{34}$$

Clearly, for any fixed $t_k \in [t_k^{\min}, t_k^{\max}]$, all the constraints (31)-(33) are linear in q_k. However, the objective function in (29) is only piecewise linear. As such, we are not faced with a standard linear program, but a piecewise linear program. Piecewise linear programs are a well-studied field of linear optimization and several techniques for solving such problems have been proposed and studied, see for instance Fourer (1985, 1988, 1992). The most straightforward approach is to solve the standard linear program corresponding to each line segment of the objective function separately, and afterward compare the solutions and store the overall optimum. This technique can be inefficient and is not recommended if the number of pieces of the objective function is relatively large. However, in our case, the objective functions normally consist of only a few pieces. For example, in the simulation experiment of Section 5.2, where a model $q(\tilde{x}|x, y)$ is constructed as much as 1,000 times, the largest number of intervals observed is 10 and the average number of intervals is 4.35. We therefore consider the straightforward approach as a convenient method for solving the piecewise linear programs in our case, but we note that more elegant strategies exist and may have their advantages. Further details of our solution are presented below.

First, some new notations needs to be introduced. For each $2 \leq k \leq n$, we let M_k denote the number of pieces, or intervals, of $E_{k:n}^*(t_k)$, and we let $t_k^{B(j)}$, $j = 1, \ldots, M_k + 1$, denote the corresponding breakpoints. Note that for the first and last breakpoints, we have $t_k^{B(1)} = t_k^{\min}$ and $t_k^{B(M_k+1)} = t_k^{\max}$. Furthermore, we let $I_k^{(j)} = [t_k^{B(j)}, t_k^{B(j+1)}] \subseteq [t_k^{\min}, t_k^{\max}]$ denote interval number j, and $\mathcal{S}_k = \{1, 2, \ldots, M_k\}$ the set of interval indices. For each $j \in \mathcal{S}_k$, $E_{k:n}^*(t_k)$ is defined by a linear function, which we denote by $E_k^{*(j)}(t_k)$, whose intercept and slope we denote by $a_k^{(j)}$ and $b_k^{(j)}$, respectively.

Each linear piece, $E_{k+1}^{*(j)}(t_{k+1})$, of the piecewise linear function $E_{(k+1):n}^*(t_{k+1})$ leads to a standard parametric linear program. Specifically, if $E_{(k+1):n}^*(t_{k+1}(t_k, q_k))$ in (29) is replaced with $E_{(k+1):n}^{*(j)}(t_{k+1}(t_k, q_k))$, we obtain an objective function

$$E_{k:n}^{(j)}(t_k, q_k) = E_k(t_k, q_k) + E_{(k+1):n}^{*(j)}(t_{k+1}(t_k, q_k)), \tag{35}$$

which is linear, not piecewise linear, as a function of q_k. The corresponding constraints for q_k are given in (31) and (32), but instead of (33), we require that t_k and q_k return a value $t_{k+1}(t_k,q_k)$ within the interval $I_{k+1}^{(j)}$,

$$t_{k+1}^{B(j)} \leq t_{k+1}(t_k, q_k) \leq t_{k+1}^{B(j+1)}. \tag{36}$$

Using (30), (34), and that $E_{k+1}^{*(j)}(t_{k+1}) = a_{k+1}^{(j)} + b_{k+1}^{(j)} t_{k+1}$, we can for each $j \in \mathcal{S}_{k+1}$ rewrite (35) as

$$E_{k:n}^{(j)}(t_k, q_k) = \beta_k^{00(j)}(t_k)q_k^{00} + \beta_k^{01(j)}(t_k)q_{n-1}^{01} + \beta_k^{10(j)}(t_k)q_k^{10} + \beta_k^{11(j)}(t_k)q_k^{11} + \alpha_k^{(j)}, \tag{37}$$

where

$$\beta_k^{00(j)}(t_k) = (b_{k+1}^{(j)}\rho_k^{0|0} + 1)\pi_k^{00}(t_k),$$

$$\beta_k^{01(j)}(t_k) = (b_{k+1}^{(j)}\rho_k^{0|1} - 1)\pi_k^{01}(t_k),$$

$$\beta_k^{10(j)}(t_k) = (b_{k+1}^{(j)}\rho_k^{0|0} + 1)\pi_k^{10}(t_k),$$

$$\beta_k^{11(j)}(t_k) = (b_{k+1}^{(j)}\rho_k^{0|1} - 1)\pi_k^{11}(t_k),$$

and

$$\alpha_k^{(j)} = a_{k+1}^{(j)} + f(x_k = 1).$$

To summarize, we obtain for each $j \in \mathcal{S}_{k+1}$ a standard parametric linear program, with the objective function given in (37) and the constraints given in (31), (32), and (36). Solving the parametric linear program corresponding to each $j \in \mathcal{S}_{k+1}$, yields the following quantities:

$$\tilde{E}_{k:n}^{(j)}(t_k) = \max_{q_k} E_{k:n}^{(j)}(t_k, q_k), \tag{38}$$

$$\tilde{q}_k^{(j)}(t_k) = \underset{q_k}{\text{argmax}}\ E_{k:n}^{(j)}(t_k, q_k). \tag{39}$$

The overall maximum value $E_{k:n}^*(t_k)$ and corresponding optimal solution $q_k^*(t_k)$ are then available as

$$E_{k:n}^*(t_k) = E_{k:n}^{j_{k+1}^*(t_k)}(t_k)$$

and

$$q_k^*(t_k) = \tilde{q}_k^{(j_{k+1}^*(t_k))}(t_k)$$

where

$$j_{k+1}^*(t_k) = \underset{j \in \mathcal{S}_{k+1}}{\text{argmax}}\ \tilde{E}_{k:n}^{(j)}(t_k).$$

As previously mentioned, and as shown in Appendix A, $E_{k:n}^*(t_k)$ is a CPL function of t_k. As such, $E_{k:n}^*(t_k)$ is fully specified by its breakpoints and the function values at those points. The breakpoints of $E_{k:n}^*(t_k)$ can be computed prior to the maximization. Thereby, we can obtain $E_{k:n}^*(t_k)$ for all values of t_k quite efficiently since we only need to solve the parametric, piecewise linear program at the breakpoints of $E_{k:n}^*(t_k)$.

Finally, consider the last step of the backward recursion, $k=1$. Here, the goal is to compute $q_{t_1}^*(\tilde{x}_1|x_1, y)$ and $E_{1:n}^*(t_1)$. Essentially, this step proceeds in the same fashion as the intermediate steps, but some technicalities are a bit different since there are only two variables involved in q_1, namely, $q_1^0 = q(\tilde{x}_1 = 0|x_1 = 0, y)$ and $q_1^1 = q(\tilde{x}_1 = 0|x_1 = 1, y)$. Also, t_1 is not a parameter free to vary within a certain range, but a fixed number, namely $t_1 = f(x_1 = 0)$, meaning that we obtain specific values for $q_{t_1}^*(\tilde{x}_1|x_1, y)$ and $E_{1:n}^*(t_1)$. The function we want to maximize at this final backward step, with respect to q_1, is

$$E_{1:n}(t_1, q_1) = E_1(t_1, q_1) + E^*_{2:n}(t_2(t_1, q_1)), \tag{40}$$

where now, recalling that $\pi(x_1|y) = f(x_1)$, the first term, $E_1(t_1,q_1)$, can be written as

$$E_1(t_1, q_1) = t_1 q_1^0 + (1 - t_1)(1 - q_1^1). \tag{41}$$

Again, as in the intermediate steps, we have a piecewise linear, not a linear, objective function. To determine the constraints for q_1, we note that the requirement (9) for $q(\tilde{x}|x, y)$ entails that

$$f(\tilde{x}_1|y) = \pi(\tilde{x}_1|y).$$

Thereby, since $t_1 = f(x_1 = 0)$ and using that $f(\tilde{x}_1|y) = \sum_{x_1} \pi(\tilde{x}_1, x_1|y)$ and $\pi(\tilde{x}_1, x_1|y) = f(x_1)q(\tilde{x}_1|x_1, y)$, we see that the following requirement must be met by $q(\tilde{x}_1|x_1, y)$:

$$f(\tilde{x}_1|y) = t_1 q(\tilde{x}_1|x_1 = 0, y) + (1 - t_1)q(\tilde{x}_1|x_1 = 1, y). \tag{42}$$

Additionally, we have the inequality constraints

$$0 \leq q_1^0, \quad q_1^1 \leq 1. \tag{43}$$

So, we are faced with a piecewise linear program, with the piecewise linear objective function (40) and the linear constraints (42) and (43). Again, we proceed by iterating through each linear piece of $E^*_{2:n}(t_2(t_1, q_1))$, solving the standard linear program corresponding to each piece separately. That is, for each $j \in \mathcal{S}_2$, we replace $E^*_{2:n}(t_2(t_1, q_1))$ in (40) by $E^{*(j)}_{2:n}(t_2(t_1, q_1))$ and consider instead the objective function

$$E^{(j)}_{1:n}(t_1, q_1) = E_1(t_1, q_1) + E^{*(j)}_{2:n}(t_2(t_1, q_1)), \tag{44}$$

which is linear, not piecewise linear, as a function of q_1. As we did for each subproblem $j \in \mathcal{S}_{k+1}$ in every intermediate backward iteration, we must for each subproblem $j \in \mathcal{S}_2$ incorporate the inequality constraints

$$t_2^{B(j)} \leq t_2(t_1, q_1) \leq t_2^{B(j+1)}, \tag{45}$$

where now $t_2(t_1,q_1)$ follows from (18) and (19) as

$$t_2(t_1, q_1) = t_1 q_1^0 \rho_1^{0|0} + (1 - t_1)q_1^1 \rho_1^{0|1}. \tag{46}$$

Using (41), (46), and that $E^{*(j)}_{2:n}(t_2) = a_2^{(j)} + b_2^{(j)} t_2$, we can rewrite the function in (44) as

$$E^{(j)}_{1:n}(t_1, q_1) = \beta_1^{0(j)}(t_1)q_1^0 + \beta_1^{1(j)}(t_1)q_1^1 + \alpha_1^{(j)}(t_1), \tag{47}$$

where

$$\beta_1^{0(j)}(t_1) = t_1(1 + b_2^{(j)} \rho_1^{0|0}),$$

$$\beta_1^{1(j)}(t_1) = (1 - t_1)(1 + b_2^{(j)} \rho_1^{0|1}),$$

$$\alpha_1^{(j)}(t_1) = 1 - t_1 + a_2^{(j)}.$$

To summarize, we obtain for each $j \in \mathcal{S}_2$ a standard linear program, where the aim is to maximize the objective function (47) with respect to q_1 subject to the constraints (42), (43), and (45). This program is solved for $t_1 = f(x_1 = 0)$. Analogously to (38) and (39), let

$$\tilde{E}_{1:n}^{(j)}(t_1) = \max_{q_1} E_{1:n}^{(j)}(t_1, q_1),$$

$$\tilde{q}_1^{(j)}(t_1) = \operatorname*{argmax}_{q_1} E_{1:n}^{(j)}(t_1, q_1).$$

Ultimately, we obtain

$$E_{1:n}^{*}(t_1) = \tilde{E}_{1:n}^{(j_2^*)}(t_1)$$

and

$$q_1^{*}(t_1) = \tilde{q}_1^{(j_2^*)}(t_1)$$

where

$$j_2^* = \operatorname*{argmax}_{j \in \mathcal{S}_2} [\tilde{E}_{1:n}^{(j)}(t_1)].$$

5 | NUMERICAL EXPERIMENTS

In this section, we demonstrate the proposed ensemble updating method for binary vectors in two simulation experiments. In Section 5.1, we present a toy example where the assumed prior $f(x)$ is a given stationary Markov chain of length $n = 4$. Here, we focus on the construction of $q(\tilde{x}|x, y)$ for this assumed prior model, not on the application of it in an ensemble-based context. In Section 5.2, we consider a higher dimensional and ensemble-based example, inspired by the movement, or flow, of water and oil in a petroleum reservoir.

5.1 | Toy example

Suppose the assumed prior $f(x)$ is a Markov chain of length $n = 4$ with homogenous transition probabilities $f(x_k = 0|x_{k-1} = 0) = 0.7$ and $f(x_k = 1|x_{k-1} = 1) = 0.8$ for $k \geq 2$, and initial distribution $f(x_1)$ equal to the associated limiting distribution. The Markov chain $f(x)$ is then a stationary chain with marginal probabilities $f(x_k = 0) = 0.40, f(x_k = 1) = 0.60$ for each $k = 1,2,3,4$. Furthermore, suppose every factor $p(y_i|x_i)$ of the likelihood model $p(y|x)$ is a Gaussian distribution with mean x_i and standard deviation $\sigma = 2$, and consider the observation vector $y = (-0.681, -1.585, 0.007, 3.103)$. The corresponding posterior Markov chain model $f(x|y)$ then

TABLE 1 Results for the optimal solution $q^*(\tilde{x}|x, y)$ of the toy example in Section 5.1, in (a) for the first factor $q^*(\tilde{x}_1|x_1, y)$, and in (b) for the remaining factors $q^*(\tilde{x}_k|x_k, \tilde{x}_{k-1}, y)$, $k = 2,3,4$

(a)		**(b)**			
k	**1**	***k***	**2**	**3**	**4**
t_k^*	0.400000	t_k^*	0.305356	0.308676	0.281108
$q_k^{*0}(t_k^*)$	1.000000	$q_k^{*00}(t_k^*)$	1.000000	1.000000	0.853968
$q_k^{*1}(t_k^*)$	0.211299	$q_k^{*01}(t_k^*)$	0.481489	0.212926	0.000000
		$q_k^{*10}(t_k^*)$	1.000000	0.860986	0.546043
		$q_k^{*11}(t_k^*)$	0.097118	0.000000	0.000000

have the transition probabilities

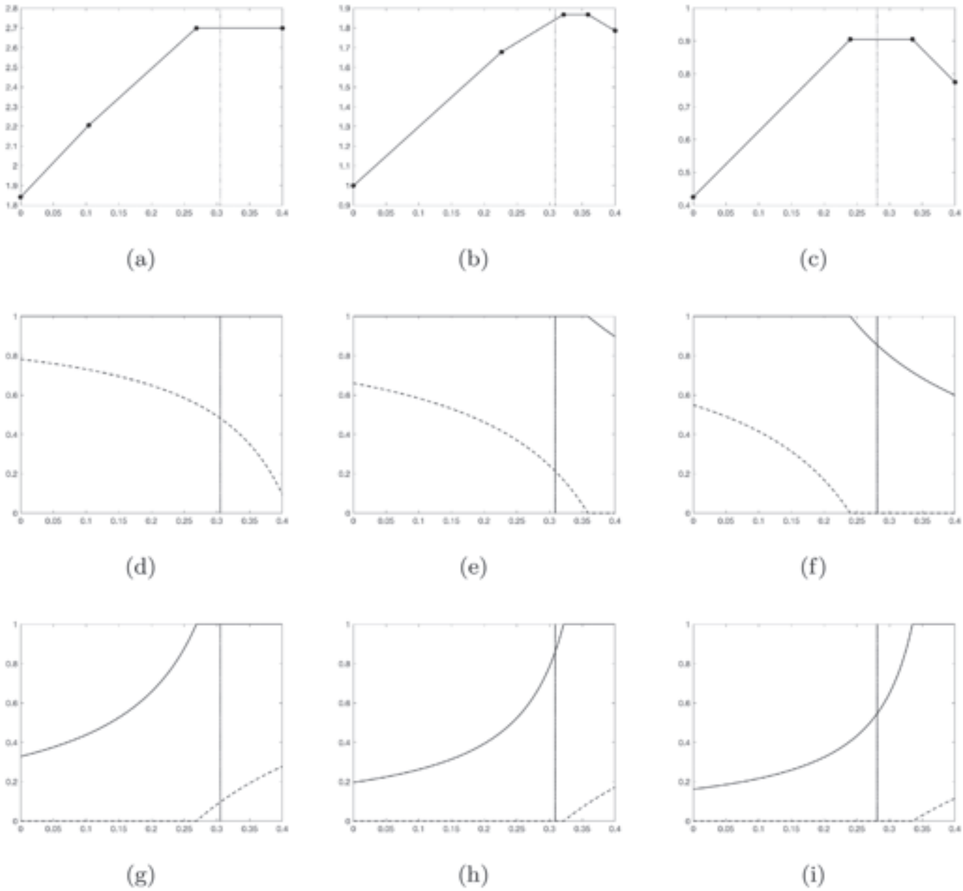
$$\begin{aligned}
&f(x_2 = 0|x_1 = 0, y) = 0.7821, \quad f(x_2 = 1|x_1 = 1, y) = 0.7223, \\
&f(x_3 = 0|x_2 = 0, y) = 0.6600, \quad f(x_3 = 1|x_2 = 1, y) = 0.8278, \\
&f(x_4 = 0|x_3 = 0, y) = 0.5490, \quad f(x_4 = 1|x_3 = 1, y) = 0.8846, \qquad (48)
\end{aligned}$$

and marginal distributions

$$\begin{aligned}
f(x_1 = 0|y) &= 0.526779, \\
f(x_2 = 0|y) &= 0.543379, \\
f(x_3 = 0|y) &= 0.437279, \\
f(x_4 = 0|y) &= 0.304977. \qquad (49)
\end{aligned}$$

Given the prior model $f(x)$ and the posterior model $f(x|y)$, we can construct $q^*(\tilde{x}|x, y)$ as described in Section 4. For this simple example, this involves computing 14 quantities, namely, $q_1^{*0}(t_1^*) = q^*(\tilde{x}_1 = 0|x_1 = 0, y)$, $q_1^{*1}(t_1^*) = q^*(\tilde{x}_1 = 0|x_1 = 1, y)$, $q_k^{*ij}(t_k^*) = q^*(\tilde{x}_k = 0|\tilde{x}_{k-1} = i, x_k = j, y)$, for $k = 2,3,4$, and $i,j = 0,1$. As described in Section 4 the construction of $q^*(\tilde{x}|x, y)$ involves a backward recursion and a forward recursion. In the backward recursion, we compute $E_{k:n}^*(t_k)$ and $q_k^{*00}(t_k)$, for $k = 4,3,2$. The results for these quantities are presented in Figure 3. In the forward recursion, we start out computing the optimal solution of the first factor, $q^*(\tilde{x}_1|x_1, y)$, and then compute the remaining optimal parameter values t_2^*, t_3^*, and t_4^* and corresponding optimal solutions $q_k^{*ij}(t_k^*)$, $k = 2,3,4$, $i,j = 0,1$. The results from the forward recursion are given in Table 1.

Taking a closer look at the results for the optimal solution $q^*(\tilde{x}|x, y)$, we see that many of the probabilities $q_k^{*ij}(t_k^*)$ are either zero or one. This feature can be formally explained mathematically (see Appendix A), but is also quite an intuitive result which has to do with how the probabilities of the prior model $f(x)$ differ from the probabilities of the posterior model $f(x|y)$. Often, if $f(x_k = 0) < f(x_k = 0|y)$, we obtain $q_k^{*00}(t_k^*) = 1$ and $q_k^{*10}(t_k^*) = 1$, while $q_k^{*01}(t_k^*)$ and $q_k^{*11}(t_k^*)$ take values somewhere between zero and one. Thus, if we have a prior sample x with $x_k = 0$, the update of x to $\tilde{x}$ is always such that $\tilde{x}_k = 0$. Specifically, in our toy example, this is the case for $k = 2$, that is, we have $f(x_2 = 0) < f(x_2 = 0|y)$, and obtained $q_2^{*00}(t_2^*) = 1$ and $q_2^{*10}(t_2^*) = 1$. Likewise, if $f(x_k = 0) > f(x_k = 0|y)$, we often obtain $q_k^{*01}(t_k^*) = 0$ and $q_k^{*11}(t_k^*) = 0$, while $q_k^{*00}(t_k^*)$ and $q_k^{*10}(t_k^*)$ take values somewhere between zero and one. Thus, if we have a prior sample x with $x_k = 1$, the



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE 3 Results from the toy example of Section 5.1. (a–c) $E^*_{k:4}(t_k)$ for $k = 2,3$, and 4, respectively, with the breakpoints highlighted as black dots. (d–f) $q_k^{*00}(t_k)$ (solid) and $q_k^{*01}(t_k)$ (dashed) for $k = 2,3$, and 4, respectively. (g–i) $q_k^{*10}(t_k)$ (solid) and $q_k^{*11}(t_k)$ (dashed) for $k = 2,3$, and 4, respectively. The vertical line in each figure represents the optimal parameter value t_k^*

update of x to $\tilde{x}$ is always such that $\tilde{x}_k = 1$. In our toy example, this is the case for $k = 4$, that is, we have $f(x_4 = 0) > f(x_4 = 0|y)$, and obtained $q_4^{*01}(t_4^*) = 0$ and $q_4^{*10}(t_4^*) = 0$. However, the model $q(\tilde{x}|x, y)$ is not only constructed so that the marginal probabilities in (49) are fulfilled, but also so that the posterior transition probabilities in (48) are reproduced. In our toy example, we see, for example, that for $k = 3$ we obtained $q_3^{*10}(t_3^*) < 1$ even if $f(x_3 = 0) < f(x_3 = 0|y)$. Instead, we observe another deterministic term, namely $q_3^{*11}(t_3^*) = 0$.

5.2 | Ensemble-based, higher dimensional example with simulated data

Until now, we have focused on the ensemble updating problem at a specific time step of the filtering recursions. However, in a practical application, one is interested in the filtering problem as a whole and needs to cope with the ensemble updating problem sequentially for $t=1,2,\dots,T$. In this section we address this issue and investigate the application of the proposed approach in this context. More specifically, we reconsider the situation with an unobserved Markov process, $\{x^t\}_{t=1}^T$, and a corresponding time series of observations, $\{y^t\}_{t=1}^T$, and at every time step $t=1,\dots,T$, we construct a distribution $q(\tilde{x}^t|x^t,y^{1:t})$ for updating the prior ensemble $\{x^{t(1)},x^{t(2)},\dots,x^{t(M)}\}$ to a posterior ensemble $\{\tilde{x}^{t(1)},\tilde{x}^{t(2)},\dots,\tilde{x}^{t(M)}\}$. Below, we first present the experimental setup of our simulation example in Section 5.2.1, and thereafter study the performance of the proposed updating approach in Sections 5.2.2 and 5.2.3.

5.2.1 | Specification of simulation example

To construct a simulation example we must first define the $\{x^t\}_{t=1}^T$ Markov process. We set $T=100$ and let $x^t=(x_1^t,\dots,x_n^t)$ be an $n=400$ dimensional vector of binary variables $x_i^t\in\{0,1\}$ for each $t=1,\dots,T$. To simplify the specification of the transition probabilities $p(x^t|x^{t-1})$ we make two Markov assumptions. First, conditioned on x^{t-1} we assume the elements in x^t to be a Markov chain so that

$$p(x^t|x^{t-1})=p(x_1^t|x^{t-1})\prod_{i=2}^{n}p(x_i^t|x_{i-1}^t,x^{t-1}).$$

The second Markov assumption we make is that

$$p(x_i^t|x_{i-1}^t,x^{t-1})=p(x_i^t|x_{i-1}^t,x_{i-1}^{t-1},x_i^{t-1},x_{i+1}^{t-1}),$$

for $i=2,\dots,n-1$, that is, the value in element i at time t only depends on the values in elements $i-1$, i, and $i+1$ at the previous time step. For $i=1$ and $i=n$ we make the corresponding Markov assumptions

$$p(x_1^t|x_1^{t-1},x_2^{t-1})\quad\text{and}\quad p(x_n^t|x_{n-1}^t,x_{n-1}^{t-1},x_n^{t-1}).$$

To specify the x^t Markov process we thereby need to specify $p(x_i^t|x_{i-1}^t,x_{i-1}^{t-1},x_i^{t-1},x_{i+1}^{t-1})$ for $t=2,\dots,T$ and $i=2,\dots,n$ and the corresponding probabilities for $t=1$ and for $i=1$ and $i=n$.

To get a reasonable test for how our proposed ensemble updating procedure works we want an $\{x^t\}_{t=1}^T$ process with a quite strong dependence between x^{t-1} and x^t, also when conditioning on observed data. Moreover, conditioned on $y^{1:t}$, the elements in x^t should not be first-order Markov so that the true model differ from the *assumed* Markov model defined in Section 3.3. In the following we first discuss the choice of $p(x_i^t|x_{i-1}^t,x_{i-1}^{t-1},x_i^{t-1},x_{i+1}^{t-1})$ for $t=2,\dots,T$ and $i=2,\dots,n$ and thereafter specify how these are modified for $t=1$ and for $i=1$ and n. When specifying the probabilities we are inspired by the process of how water comes through to an oil producing well in a petroleum reservoir, but without claiming our model to be a very realistic model for this situation. Thereby t represents time and i the location in the well. We let $x_i^t=0$ represent the presence of oil at location or node i at time t and correspondingly $x_i^t=1$ represents the presence of water. In the start we assume oil is present in the whole well, but as time goes by more and more water is present and at time $t=T$ water has become the dominating fluid in the well. Whenever $x_i^{t-1}=1$ we therefore want $x_i^t=1$ with very high probability, especially if also $x_{i-1}^t=1$. If $x_i^{t-1}=0$ we correspondingly want a high probability for $x_i^t=0$ unless $x_{i-1}^t=1$ and $x_{i-1}^{t-1}=x_{i+1}^{t-1}=1$. Trying different

TABLE 2 Probabilities defining the true model $p(x^t|x^{t-1})$ used to simulate a true chain $\{x^t\}_{t=1}^T$ in the simulation experiment presented in Section 5.2

$\mathbf{x_{i-1}^{t-1}}$	$\mathbf{x_{i}^{t-1}}$	$\mathbf{x_{i+1}^{t-1}}$	$\mathbf{p(x_i^t = 1 \vert x_{i-1}^t = 1, x_{i-1:i+1}^{t-1})}$	$\mathbf{p(x_i^t = 1 \vert x_{i-1}^t = 0, x_{i-1:i+1}^{t-1})}$
0	0	0	.0100	.0050
1	0	0	.0400	.0100
0	1	0	.9999	.9800
1	1	0	.9999	.9900
0	0	1	.0400	.0400
1	0	1	.9800	.0400
0	1	1	.9999	.9800
1	1	1	.9999	.9800

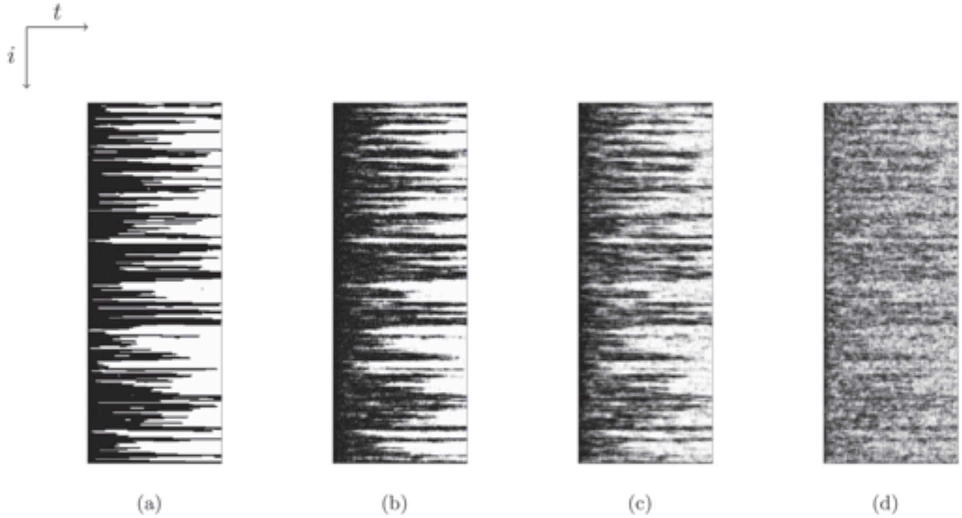


FIGURE 4 Results from the simulation experiment of Section 5.2: Grayscale images of (a) the unobserved process $\{x^t\}_{t=1}$, (b) $\{\hat{p}_c(x_i^t|y^{1:t})\}_{t=1}^{100}$, (c) $\{\hat{p}_q(x_i^t|y^{1:t})\}_{t=1}^{100}$, and (d) $\{\hat{p}_a(x_i^t|y^{1:t})\}_{t=1}^{100}$. The colors black and white correspond to the values zero and one, respectively

sets of parameter values according to these rules we found that the values specified in Table 2 gave realizations consistent with the requirements discussed above. One realization from this model is shown in Figure 4a, where black and white represent 0 (oil) and 1 (water), respectively. The corresponding probabilities when $t = 1$ and for $i = 1$ and n we simply define from the values in Table 2 by defining all values lying outside the $\{(i,t):i = 1, \dots ,n;t = 1, \dots ,T\}$ lattice to be zero. In particular this implies that at time $t = 0$, which is outside the lattice, oil is present in the whole well. In the following we consider the realization shown in Figure 4a to be the (unknown) true x^t process.

The next step in specifying the simulation example is to specify an observational process. For this we simply assume one scalar observation y_i^t for each node i at each time t, and assume the elements in $y^t = (y_1^t, \dots ,y_n^t)$ to be conditionally independent given x^t. Furthermore, we let y_i^t be

Gaussian with mean x_i^t and variance σ^2. As we want the dependence between x^{t-1} and x^t to be quite strong also when conditioning on the observations, we need to choose the variance σ^2 reasonably large, so we set $\sigma^2 = 2^2$. Given the true x^t process shown in Figure 4a we simulate y_i^t values from the specified Gaussian distribution, and in the following consider these values as observations. An image of these observations is not included, since the variance σ^2 is so high that such an image is not very informative.

Pretending that the $\{x^t\}_{t=1}^T$ process is unknown and that we only have the observations $\{y^t\}_{t=1}^T$ available, our aim with this simulation study is to evaluate how well our proposed ensemble based filtering procedure is able to capture the properties of the correct filtering distributions $p(x^t|y^{1:t}), t = 1, \dots , T$. To do so we first need to evaluate the properties of the correct filtering distributions. It is possible to get samples from $p(x^t|y^{1:t})$ by simulating from $p(x^{1:t}|y^{1:t})$ with a Metropolis–Hastings algorithm, but to a very high computational cost as a separate Metropolis–Hastings run must be performed for each value of t. Nevertheless, we do this to get the optimal solution of the filtering problems to which we can compare the results of our proposed ensemble based filtering procedure. In our algorithm for simulating from $p(x^{1:t}|y^{1:t})$ we combine single site Gibbs updates of each element in $x^{1:t}$ with a one-block Metropolis–Hastings update of all elements in $x^{1:t}$. To get a reasonable acceptance rate for the one-block proposals we adopt the approximation procedure introduced in Austad and Tjelmeland (2017) to obtain a partially ordered Markov model (Cressie & Davidson, 1998) approximation to $p(x^{1:t}|y^{1:t})$, propose potential new values for $x^{1:t}$ from this approximate posterior, and accept or reject the proposed values according to the usual Metropolis–Hastings acceptance probability. For each value of t we run the Metropolis–Hastings algorithm for a large number of iterations and discard a burn-in period. From the generated realizations we can then estimate the properties of $p(x^t|y^{1:t})$. In particular we can estimate the marginal probabilities $p(x_i^t = 1|y^{1:t})$ as the fraction of realizations with $x_i^t = 1$. We denote these estimates of the correct filtering probabilities by $\hat{p}_c(x_i^t = 1|y^{1:t})$. In Figure 4b all these estimates are visualized as a grayscale image, where black and white correspond to $\hat{p}_c(x_i^t = 1|y^{1:t})$ equal to zero and one, respectively. It is important to note that Figure 4b is not showing the solution of the smoothing problem, but the solution of many filtering problems put together as one image.

Given the simulated observations $\{y^t\}_{t=1}^{100}$ and the model specifications described above, the proposed ensemble filtering method is run using the ensemble size $M = 20$. This is quite a small ensemble size compared with $n = 400$. The reason for choosing the ensemble size this small is to keep the simulation experiment as realistic as possible, and in real-world problems it is often necessary to set M rather small for computational reasons. A problem, however, when the ensemble size is this small compared with n, is that results may vary a lot from one run to another. To quantify this between-run variability, we therefore rerun the proposed approach a total of $B = 1{,}000$ times, each time with a new initial ensemble of $M = 20$ realizations from the initial model $p(x^1)$. At each time step t we thus achieve a total of $MB = 20{,}000$ posterior samples of the state vector x^t which can be used to construct an estimate, denoted $\hat{p}_q(x^t|y^{1:t})$, for the true filtering distribution $p(x^t|y^{1:t})$.

An important step of the proposed approach is the estimation of a first-order Markov chain $f(x^t|y^{1:t-1})$ at each time step t. Basically, this involves estimating an initial distribution $f(x_1^t|y^{1:t-1})$ and $n-1$ transition matrices $f(x_{i+1}^t|x_i^t, y^{1:t-1}), i = 1, \dots , n-1$. Since each component x_i^t is a binary variable, the initial distribution $f(x_1^t|y^{1:t-1})$ can be represented by one parameter, while the transition matrices $f(x_{i+1}^t|x_i^t, y^{1:t-1})$ each require two parameters. In this example, we pursue a Bayesian approach for estimating these parameters. Specifically, if we let θ^t represent a vector containing all the parameters required to specify the model $f(x^t|y^{1:t-1})$, we put a prior on θ^t, $f(\theta^t)$, and then set the final estimator for θ^t equal to the mean of the corresponding posterior distribution $f(\theta^t|x^{t,(1)}, \dots, x^{t,(M)})$. In the specification of $f(\theta^t)$ we assume that all the parameters in the vector θ^t are independent and that each parameter follows a Beta distribution $\mathcal{B}(\alpha, \beta)$ with known hyperparameters $\alpha = 2, \beta = 2$.

To get a better understanding of the performance of the proposed approach, we also implement another, more naïve procedure to which our results can be compared. The naïve procedure is essentially the same as the proposed approach but at each time step t we do not construct a $q(\tilde{x}^t|x^t, y^{1:t})$ and instead update the prior ensemble by simulating independent samples from the assumed Markov chain model $f(x^t|y^{1:t})$. Below, we refer to this method as the assumed model approach. As with the proposed approach, we rerun the assumed model approach $B = 1{,}000$ times. This yields a total of $MB = 20{,}000$ posterior samples of each state vector x^t, $t = 1, \dots, T$, which can be used to construct an estimate, denoted $\hat{p}_a(x^t|y^{1:t})$, for the true filtering distribution $p(x^t|y^{1:t})$. By comparing $\hat{p}_a(x^t|y^{1:t})$ and $\hat{p}_q(x^t|y^{1:t})$ with the MCMC estimate $\hat{p}_c(x^t|y^{1:t})$, which essentially represents the true model $p(x^t|y^{1:t})$, we can get an understanding of how much we gain by executing the proposed approach instead of the much simpler assumed model approach. In the next two sections we investigate how well $\hat{p}_q(x^t|y^{1:t})$ and $\hat{p}_a(x^t|y^{1:t})$ capture marginal and joint properties of the true distribution $p(x^t|y^{1:t})$ for which the MCMC estimate $\hat{p}_c(x^t|y^{1:t})$ works as a reference.

Before we present our results, we mention that we also tried to implement the method of Oliver et al. (2011). This method has the advantage of being relatively easy to implement and slightly less computer-demanding than the proposed approach. However, we could not obtain useful results with this method when the ensemble size was as small as $M = 20$. For simplicity, the results are therefore not included in the next sections. We note, however, that the results obtained with larger ensemble sizes were more promising. In our implementation of the algorithm, we used a first-order Markov chain as the prior model, and to estimate this Markov chain we used the Bayesian procedure described above, that is, the same procedure that was used to estimate the first-order Markov chain at every time step in the two other updating methods. Perhaps using a higher order Markov chain, which indeed is possible in the method of Oliver et al. (2011), could help to improve the results for the small ensemble size $M = 20$. Moreover, we only applied a basic EnKF in our implementation. It is possible that using a more advanced EnKF scheme which for example incorporates inflation and/or localization could improve the results.

5.2.2 | Evaluation of marginal distributions

In this section, we are interested in studying how well the proposed approach estimates the marginal filtering distributions $p(x_i^t|y^{1:t})$, $i = 1, \dots, n$, $t = 1, \dots, T$. Following the notations introduced above, we let $\hat{p}_q(x_i^t|y^{1:t})$ and $\hat{p}_a(x_i^t|y^{1:t})$ denote estimates of the marginal distribution $p(x_i^t|y^{1:t})$ obtained with the proposed approach and the assumed model approach, respectively. The values of $\hat{p}_q(x_i^t = 1|y^{1:t})$ and $\hat{p}_a(x_i^t = 1|y^{1:t})$ are in each case set equal to the mean of the corresponding set of samples of x_i^t. Figure 4c,d presents grayscale images of $\{\hat{p}_q(x_i^t = 1|y^{1:t})\}_{t=1}^{100}$ and $\{\hat{p}_a(x_i^t = 1|y^{1:t})\}_{t=1}^{100}$, respectively. From a visual inspection, the image of $\{\hat{p}_q(x_i^t = 1|y^{1:t})\}_{t=1}^{100}$ is more gray and noisy than that of $\{\hat{p}_c(x_i^t = 1|y^{1:t})\}_{t=1}^{100}$ shown in Figure 4b which contains more tones closer to pure black and white. This is to be expected, since $\{\hat{p}_c(x_i^t|y^{1:t})\}_{t=1}^{100}$ essentially is the ideal solution, and we cannot expect an approximate method to perform this well. However, the image of $\{\hat{p}_a(x_i^t = 1|y^{1:t})\}_{t=1}^{100}$ is even more gray and noisy than that of $\{\hat{p}_q(x_i^t = 1|y^{1:t})\}_{t=1}^{100}$, so it seems that we do gain something by running the proposed approach instead of the simpler assumed model approach. To investigate this further, we compute the Frobenius norms of the two matrices obtained by subtracting the true marginal probabilities $\hat{p}_c(x_i^t = 1|y^{1:t})$ from the corresponding estimates $\hat{p}_q(x_i^t = 1|y^{1:t})$ and $\hat{p}_a(x_i^t = 1|y^{1:t})$. We then obtain the numbers 35.38 and 63.00, respectively. That is, the Frobenius norm of the difference between the true and the estimated marginal filtering distributions is reduced to almost the half with the proposed approach compared with the assumed model approach. This clearly suggests that we overall obtain much better estimates of the marginal distributions $p(x_i^t|y^{1:t})$ with the proposed method than with the assumed model approach.

To look further into the accuracy of the marginal estimates $\hat{p}_q(x_i^t = 1|y^{1:t})$ and $\hat{p}_a(x_i^t = 1|y^{1:t})$ and to study their variability from run to run, we take a closer look at the results for some specific time steps. For each of these time steps we compute a 90% quantile interval for each of the estimates $\hat{p}_q(x_i^t = 1|y^{1:t})$ and $\hat{p}_a(x_i^t = 1|y^{1:t})$, $i = 1, \dots, 400$. To compute the quantile intervals, recall that the proposed approach and the assumed model approach were both rerun $B = 1{,}000$ times. This means that from each run $b = 1, \dots, B$ of the proposed approach, we have an estimate $\hat{p}_q^{(b)}(x_i^t|y^{1:t})$ of $p(x_i^t|y^{1:t})$ for each i. Likewise, from each run $b = 1, \dots, B$ of the assumed model approach, we have an estimate $\hat{p}_a^{(b)}(x_i^t|y^{1:t})$ of $p(x_i^t|y^{1:t})$ for each i. Hence, for each marginal distribution $p(x_i^t|y^{1:t})$, we have $B = 1{,}000$ estimates $\{\hat{p}_q^{(b)}(x_i^t|y^{1:t})\}_{b=1}^B$ obtained with the proposed approach and $B = 1{,}000$ estimates $\{\hat{p}_a^{(b)}(x_i^t|y^{1:t})\}_{b=1}^B$ obtained with the assumed model approach. From these two sets of samples, corresponding quantile intervals for $\hat{p}_q(x_i^t = 1|y^{1:t})$ and $\hat{p}_a(x_i^t = 1|y^{1:t})$ can be constructed. Figure 5 presents the computed results for time step $t = 60$. For simplicity, we do not include corresponding figures from the other time steps that we studied, since they look very much the same as those obtained for time $t = 60$. According to Figure 5a,b, it seems that the essentially true value $\hat{p}_c(x_i^{60}|y^{1:60})$ typically lies within the 90% quantile interval corresponding to $\hat{p}_q(x_i^{60}|y^{1:60})$, but often closer to one of the interval boundaries rather than the estimate $\hat{p}_q(x_i^{60}|y^{1:60})$ itself. In particular, we note that $\hat{p}_c(x_i^{60}|y^{1:60})$ often is close to either zero or one, while $\hat{p}_q(x_i^{60}|y^{1:60})$ is a bit higher than zero or a bit lower than one. This is not unreasonable, since we have used approximations to construct $\hat{p}_q(x_i^{60}|y^{1:60})$. Thereby, we loose information about the true quantity $\hat{p}_c(x_i^{60}|y^{1:60})$ and end up with estimated values closer to 0.5. From Figure 5c,d, we observe that this is even more the case for the estimate $\hat{p}_a(x_i^{60}|y^{1:60})$ whose quantile interval often not even covers $\hat{p}_c(x_i^{60}|y^{1:60})$.

5.2.3 | Evaluation of joint distributions

In this section, we want to evaluate how well the proposed approach manages to capture properties about the joint distribution $p(x^t|y^{1:t})$. To do so, we select three specific time steps to study, namely $t = 60$, $t = 70$, and $t = 80$. For each of these steps, we perform two tests on our samples, both concerning a feature we refer to as *contact* between a pair of nodes of x^t. Consider two components x_i^t and x_j^t of x^t at a given time step t. Given that x_i^t is equal to one, that is, $x_i^t = 1$, we say that there is contact between node i and node j in x^t if all components of x^t between and including node i and node j are equal to one. That is, there is contact between node i and j, given that x_i^t is equal to one, if the function

$$\kappa_{ij}(x^t) = \begin{cases} 1(x_j^t = 1 \cap x_{j+1}^t \cap \dots \cap x_i^t = 1), & \text{if } j \le i, \\ 1(x_i^t = 1 \cap x_{i+1}^t \cap \dots \cap x_j^t = 1), & \text{if } j > i, \end{cases}$$

is equal to one.

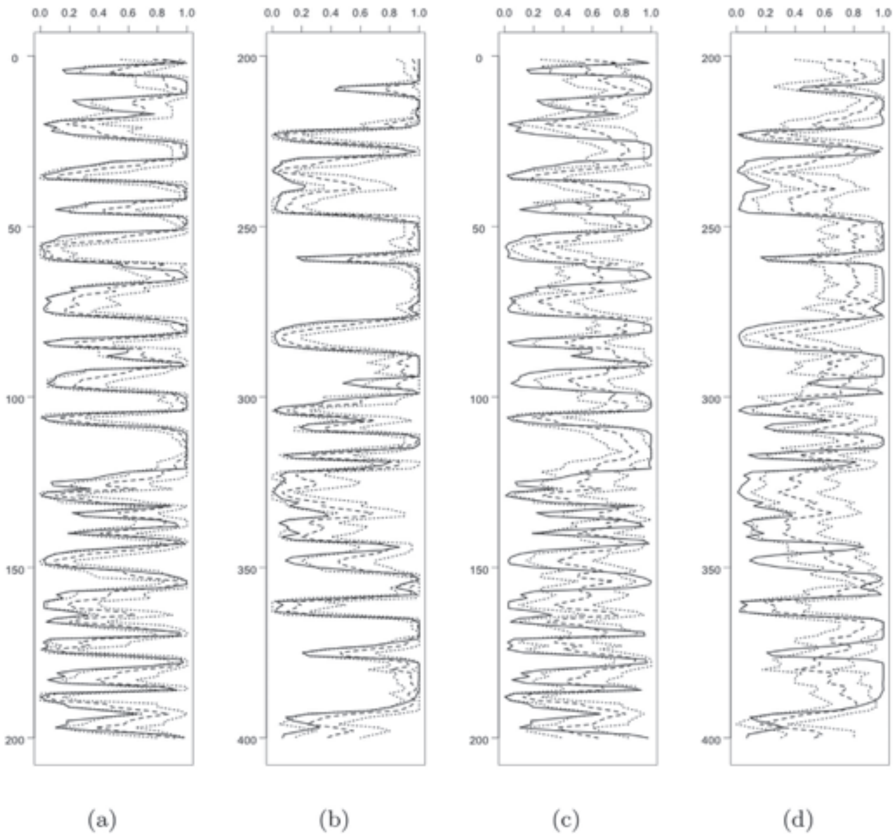


FIGURE 5 Results obtained at time step $t = 60$ in the numerical experiment of Section 5.2. (a,b) Marginal estimates $\hat{p}_q(x_i^t = 1|y^{1:t})$ (dashed) and corresponding 90% quantile intervals (dotted), in (a) from $i = 1$ to $i = 200$, and in (b) from $i = 201$ to $i = 400$. (c,d) Corresponding results for $\hat{p}_a(x_i^t = 1|y^{1:t})$. The solid line in each plot represent the MCMC estimate $\hat{p}_c(x_i^t|y^{1:t})$

Keeping i fixed, we are in our first test interested in studying the probability that there is contact between node i and node j for various values of j, given that x_i^t is equal to one. Mathematically, that means we are interested in

$$p^t(i,j) = \text{Prob}(\kappa_{ij}(x^t) = 1|x_i^t = 1, y^{1:t}). \tag{50}$$

It is most informative to study (50) for a node i whose corresponding component x_i^t has a high probability of being equal to one. Therefore, we concentrate on estimating (50) for three specific choices of i, each corresponding to a component x_i^t with a relatively high probability of being equal to one. According to the grayscale images in Figure 4 this appears to be the case for the three nodes $i = 115$, $i = 210$, and $i = 290$ at all three time steps $t = 60$, $t = 70$, and $t = 80$. For each i and t, we can then use our three sets of samples of x^t to obtain three different estimates of (50) for all j. Following previous notations, we let $\hat{p}_c^t(i,j)$ denote the MCMC estimate of $p^t(i,j)$, while $\hat{p}_q^t(i,j)$ and $\hat{p}_a^t(i,j)$ denote the estimates obtained with the proposed approach and the assumed model

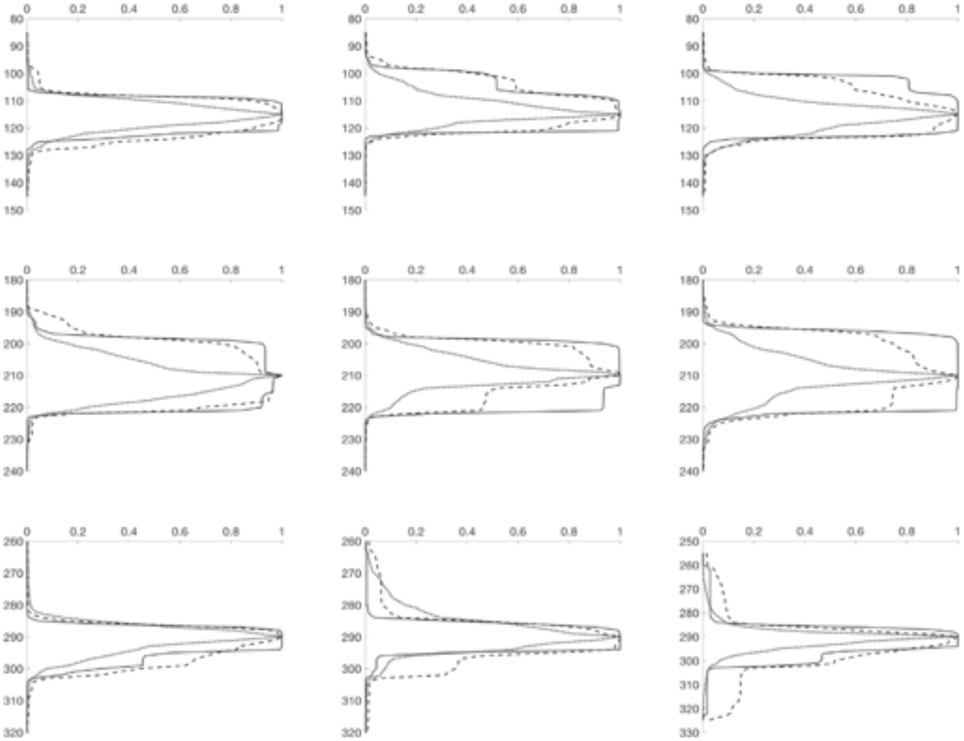


FIGURE 6 Results from the numerical experiment of Section 5.2. The graphs present $\hat{p}_c^t(i,j)$ (solid), $\hat{p}_q(i,j)$ (dashed), and $\hat{p}_a(i,j)$ (dotted) for the three components $i = 115$, $i = 210$, and $i = 290$ at time steps $t = 60$ (left column), $t = 70$ (middle column), and $t = 80$ (right column)

approach, respectively. Figure 6 presents the computed results. Comparing the curves representing the estimates $\hat{p}_c^t(i,j)$, $\hat{p}_q^t(i,j)$, and $\hat{p}_a^t(i,j)$, we observe that $\hat{p}_q^t(i,j)$ and $\hat{p}_a^t(i,j)$ typically decrease to zero for increasing values of j quicker than $\hat{p}_c^t(i,j)$ does. However, we see that $\hat{p}_a^t(i,j)$ decreases considerably faster than $\hat{p}_q^t(i,j)$. This makes sense, since the posterior samples used to construct the estimate $\hat{p}_a^t(i,j)$ are drawn independently from the assumed model $f(x^t|y^{1:t})$, not taking the state of the prior samples into account.

In our second test, we focus on the total number of nodes an arbitrary node i with $x_i^t = 1$ is in contact with. We denote this quantity by $L_i(x^t)$. Mathematically, $L_i(x^t)$ can be written

$$L_i(x^t) = \max_{j \geq i} \{j; \kappa_{ij}(x^t) = 1\} - \min_{j \leq i} \{j; \kappa_{ij}(x^t) = 1\} + 1.$$

For each of the time steps $t = 60$, $t = 70$, and $t = 80$, we want to study the cumulative distribution of $L_i(x^t)$,

$$F(l) = \text{Prob}(L_i(x^t) \leq l | x_i^t = 1), \tag{51}$$

when randomizing over both i and x^t, with $i \sim \text{unif}\{1,n\}$ and $x^t \sim p(x^t|y^{1:t})$. Again, we can use our three sets of samples to construct three different estimates of (51). That is, we can construct

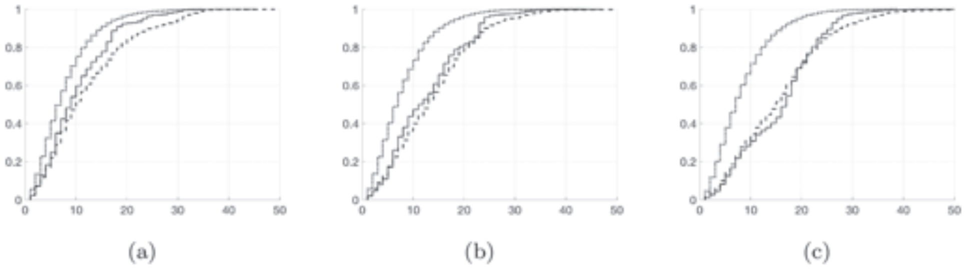


FIGURE 7 Results from the numerical experiment of Section 5.2. Estimates of $F(l) = P(L_i(x^t) \leq l | x_i^t = 1)$ with $i \sim \text{unif}\{1,n\}$ and $x^t \sim p(x^t|y^{1:t})$. The graphs present $\hat{F}_c(l)$ (solid), $\hat{F}_q(l)$ (dashed), and $\hat{F}_a(l)$ (dotted) at time steps (a) $t = 60$, (b) $t = 70$, and (c) $t = 80$

$\hat{F}_c(l)$ from the MCMC samples, $\hat{F}_q(l)$ from the samples generated with the proposed approach, and $\hat{F}_a(l)$ from the samples generated with the assumed model approach. Figure 7 presents the results. Here, we see that $\hat{F}_a(l)$ is above $\hat{F}_c(l)$ at all three time steps $t = 60$, 70, and 80, indicating that $L_i(x^t)$ typically is too small and that the assumed model approach underestimates the level of contact between nodes. This makes sense and agrees with the behavior of $\hat{p}_a^t(i,j)$ discussed above. According to Figure 7b,c, the estimate $\hat{F}_q(l)$ obtained with the proposed approach appears to do a better job since it is relatively close to $\hat{F}_c(l)$. We note, however, that this is not the case in Figure 7a; here, the curve for $\hat{F}_q(l)$ is below $\hat{F}_c(l)$, suggesting that $L_i(x^t)$ typically is too high. To investigate this further we also examined corresponding output from other time steps t. We then observed that for smaller values of t, typically smaller than 60, the curve for $\hat{F}_q(l)$ tends to be below $\hat{F}_c(l)$, while for larger values of t, it tends to be quite close to $\hat{F}_c(l)$. This is in fact not so unreasonable, since it is for higher values of t that the value one (i.e., water) is dominant in x^t. For smaller values of t, the value zero (i.e., oil) becomes more and more dominant, and the length of one-valued chains is not supposed to be very high. Perhaps our optimality criterion of maximizing the expected number of unchanged components in this case results in keeping too much information from the prior samples.

6 | CLOSING REMARKS

An approximate and ensemble-based method for solving the filtering problem is presented. The method is particularly designed for binary state vectors and is based on a generalized view of the well-known EnKF. In the EnKF, a Gaussian approximation $f(x)$ to the true prior is constructed which combined with a linear-Gaussian likelihood model yields a Gaussian approximation $f(x|y)$ to the true posterior. The prior ensemble is then updated with a linear shift such that the distribution of each updated sample is equal to $f(x|y)$ provided that the distribution of the prior samples is equal to $f(x)$. In the proposed approach for binary vectors we instead choose $f(x)$ as a first-order Markov chain. Combined with a particular likelihood model, a corresponding posterior Markov chain $f(x|y)$ can be computed. To update the prior samples, we construct a distribution $q(\tilde{x}|x,y)$ and simulate the updated samples from this distribution. Similarly to the EnKF, we want to construct $q(\tilde{x}|x,y)$ so that the updated samples are distributed according to $f(x|y)$ given that the prior samples are distributed according to $f(x)$. However, constructing such a $q(\tilde{x}|x,y)$ different from $f(x|y)$ itself is generally too intricate and we therefore consider an approximate solution.

Specifically, instead of requiring that $q(\tilde{x}|x, y)$ retains the Markov chain model $f(x|y)$ exactly, we require only that it retains all the marginal distributions $f(x_i,x_{i+1}|y)$, $i = 1, \dots n-1$. Based on the optimality criterion of maximizing the expected number of unchanged components, an optimal solution of $q(\tilde{x}|x, y)$ is computed with dynamic programming techniques. According to the results from a simulation experiment, the performance of the proposed updating method is promising.

The focus of this article is on binary state vectors with a one-dimensional spatial arrangement. Clearly, this is a simple situation with limited practical interest since most real problems involve at least two spatial dimensions and multiple classes for the state variables. Nevertheless, we consider the work of this article as a first step toward a more advanced method, and in the future we would like to explore possible extensions of the proposed method. Conceptually, most of the material presented in the article can easily be generalized to more complicated situations. Computationally, however, it is more challenging. A generalization of the material in Sections 3 and 4 to a similar situation with more than two possible classes, involves a growing number of free parameters in the construction of each factor $q(\tilde{x}_k|\tilde{x}_{k-1}, x_k, y)$. Specifically, in the case of three classes there will be four parameters involved, while in the case of four classes there will be nine parameters involved. We believe, however, that it is possible to cope with a situation with more than one free parameter via an iterative procedure. Specifically, one can start with some initial values for each of the free parameters and thereafter iteratively optimize with respect to one of the parameters at a time, keeping the other parameters fixed. By iterating until convergence we thereby obtain the optimal solution. How many parameters we are able to deal with using this strategy will depend on how fast convergence is reached and, of course, how much computation time one is willing to use.

Another possible extension of our method is to pursue a higher order Markov chain for the assumed prior model $f(x)$. If this is possible, a further generalization to two spatial dimensions may be possible by choosing a Markov mesh model (Abend, Harley, & Kanal, 1965) for $f(x)$. Being able to cope with higher order Markov models will also allow the use of more complicated likelihood models where, for example, each observation is a function of several x_i's. However, similarly to the case with multiple classes, the computational complexity grows rapidly with the order of the Markov chain. The higher the order, the higher the number of free parameters there will be in the construction of each factor $q(\tilde{x}_k|\tilde{x}_{k-1}, x_k, y)$. Computationally we can again imagine to cope with this situation by adopting an iterative optimization algorithm as discussed above.

An optimality criterion needs to be specified when constructing $q(\tilde{x}|x, y)$. In our work we choose to define the optimal solution as the one that maximizes the expected number of equal components. To us this seems like an intuitively reasonable criterion, since we want to retain as much information as possible from the prior samples. However, there may be other criteria that are more suitable and which might improve the performance of our procedure. What optimality criterion that gives the best results may even depend on how the true and assumed distributions differ. One may therefore imagine to construct a procedure which at each time t use the prior samples to estimate, or select, the best optimality criterion within a specified class.

In the future, we would also like to investigate more thoroughly the EnKF and its part within the proposed ensemble updating framework. In the present article, we impose an optimality criterion for the updating of a binary state vector, but do not focus on appropriate optimality conditions in the EnKF. For the square root filter, the matrix B in the linear update (5) is not unique except in the univariate case, which gives rise to a class of square root algorithms. It would be interesting to investigate the solution of B under different optimality conditions. One possible criterion is a continuous equivalent to the optimality criterion considered in the binary case, namely, to minimize the expected change between a prior and posterior state vector. For the stochastic EnKF, the situation is different. Here, there is no flexibility and the filter is already optimal in some sense. It is, however, not straightforward to understand specifically what the corresponding optimality criterion is.

ORCID

Margrethe Kvale Loe https://orcid.org/0000-0003-1357-9173

REFERENCES

Abend, K., Harley, T., & Kanal, L. (1965). Classification of binary random patterns. *IEEE Transactions on Information Theory*, *11*(4), 538–544.

Anderson, J. L. (2001). An ensemble adjustment Kalman filter for data assimilation. *Monthly Weather Review*, *129*(12), 2884–2903.

Anderson, J. L. (2007a). An adaptive covariance inflation error correction algorithm for ensemble filters. *Tellus A: Dynamic Meteorology and Oceanography*, *59*(2), 210–224.

Anderson, J. L. (2007b). Exploring the need for localization in ensemble data assimilation using a hierarchical ensemble filter. *Physica D: Nonlinear Phenomena*, *230*(1), 99–111.

Austad, H. M., & Tjelmeland, H. (2017). Approximate computations for binary Markov random fields and their use in Bayesian models. *Statistics and Computing*, *27*(5), 1271–1292.

Bishop, C. H., Etherton, B. J., & Majumdar, S. J. (2001). Adaptive sampling with the ensemble transform Kalman filter. Part 1: Theoretical aspects. *Monthly Weather Review*, *129*(3), 420–436.

Burgers, G., van Leeuwen, P. J., & Evensen, G. (1998). Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, *126*(6), 1719–1724.

Cressie, N., & Davidson, J. (1998). Image analysis with partially ordered Markov models. *Computational Statistics & Data Analysis*, *29*(1), 1–26.

Doucet, A., de Freitas, N., & Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. New York, NY: Springer-Verlag.

Evensen, G. (1994). Sequential data assimilation with a non-linear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Geophysical Research*, *99*(C5), 10143–10162.

Evensen, G. (2009). *Data assimilation: The ensemble Kalman filter*. Berlin, Germany: Springer Science & Business Media.

Fourer, R. (1985). A simplex algorithm for piecewise-linear programming I: Derivation and proof. *Mathematical Programming*, *33*(2), 204–233.

Fourer, R. (1988). A simplex algorithm for piecewise-linear programming II: Finiteness, feasibility and degeneracy. *Mathematical Programming*, *43*(1-3), 281–315.

Fourer, R. (1992). A simplex algorithm for piecewise-linear programming III: Computational analysis and applications. *Mathematical Programming*, *53*(3), 213–235.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, *82*, 35–45.

Künsch, H. (2000). *State space and hidden Markov models*. In O. Barndorff-Nielsen & C. Kluppelberg (Eds.), *Complex stochastic systems* (pp. 109–174). Boca Raton, FL: Chapman & Hall/CRC.

Oliver, D. S., Chen, Y., & Nævdal, G. (2011). Updating Markov Chain models using the ensemble Kalman filter. *Computational Geosciences*, *15*(2), 325–344.

Sætrom, J., & Omre, H. (2013). Uncertainty quantification in the ensemble Kalman filter. *Scandinavian Journal of Statistics*, *40*(4), 868–885.

Tippett, M. K., Anderson, J. L., Bishop, C. H., & Hamill, T. M. (2003). Ensemble square root filters. *Monthly Weather Review*, *131*(7), 1485–1490.

van Leeuwen, P. J. (2010). Nonlinear data assimilation in geosciences: An extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*, *136*, 1001–1999.

van Leeuwen, P. J. (2011). Efficient nonlinear data-assimilation in geophysical fluid dynamics. *Computers & Fluids*, *46*, 52–58.

Villani, C. (2009). *Optimal transport*. Berlin Heidelberg/Germany: Springer-Verlag.

Whitaker, J. S., & Hamill, T. M. (2002). Ensemble data assimilation without perturbed observations. *Monthly Weather Review*, *130*(7), 1913–1924.

How to cite this article: Loe MK, Tjelmeland H. Ensemble updating of binary state vectors by maximizing the expected number of unchanged components. *Scand J Statist*. 2020;1–38. https://doi.org/10.1111/sjos.12483

APPENDIX A

This appendix provides an informal proof of that $E^*_{k:n}(t_k)$, $2 \leq k \leq n$, is continuous piecewise linear (CPL). Every iteration of the backward recursion, except the first, relies on this result. The proof is an induction proof and consists of two main steps. First, in Section A.1, we consider the first step of the backward recursion and prove that $E^*_n(t_n)$ is CPL. This corresponds to the "base case" of our induction proof. Next, in Section A2, we consider the intermediate steps and prove that $E^*_{k:n}(t_k)$ is also CPL, given that $E^*_{k+1:n}(t_{k+1})$ is CPL, $2 \leq k < n$. This corresponds to the "inductive step" of our induction proof. In Section A3 of the appendix, we explain how to determine the breakpoints of $E^*_{k:n}(t_k)$, $2 \leq k < n$, prior to solving the corresponding parametric, piecewise linear program. This is crucial in order to avoid a numerical computation of $E^*_{k:n}(t_k)$ on a grid of t_k-values. Throughout the appendix, we assume the reader is familiar with all notations introduced in the previous sections of the article.

The first iteration

The parametric linear program of the first backward iteration can easily be computed analytically. Because of the equality constraints in (27) we can reformulate the optimization problem in terms of two variables instead of four. More specifically, we can choose either q_n^{00} or q_n^{01} from (27a), together with either q_n^{10} or q_n^{11} from (27b), and then reformulate the problem in terms of the two chosen variables. Here, we choose q_n^{00} and q_n^{10}. By rearranging terms in (27a) and (27b) we can write

$$\pi_n^{01}(t_n)q_n^{01} = f_n^{00} - \pi_n^{00}(t_n)q_n^{00}, \tag{A1}$$

$$\pi_n^{11}(t_n)q_n^{11} = f_n^{10} - \pi_n^{10}(t_n)q_n^{10}. \tag{A2}$$

Now, if we replace the terms $\pi_n^{01}(t_n)q_n^{01}$ and $\pi_n^{11}(t_n)q_n^{11}$ in the objective function $E_n(t_n,q_n)$ in (26) with the right-hand side expressions in (A1) and (A2), respectively, we can rewrite $E_n(t_n,q_n)$ in terms of q_n^{00} and q_n^{10} as

$$E_n(t_n, q_n) = 2\pi_n^{00}(t_n)q_n^{00} + 2\pi_n^{10}(t_n)q_n^{10} + c_n, \tag{A3}$$

where c_n is a constant given as

$$c_n = f(x_n = 1) - f(x_n = 0|y).$$

Furthermore, combining (A1) and (A2) with the inequality constraints (28) allows us to reformulate the constraints for q_n^{00} and q_n^{10} as

$$\max\left\{0, \frac{f_n^{00} - \pi_n^{01}(t_n)}{\pi_n^{00}(t_n)}\right\} \le q_n^{00} \le \min\left\{1, \frac{f_n^{00}}{\pi_n^{00}(t_n)}\right\}, \tag{A4}$$

$$\max\left\{0, \frac{f_n^{10} - \pi_n^{11}(t_n)}{\pi_n^{10}(t_n)}\right\} \le q_n^{10} \le \min\left\{1, \frac{f_n^{10}}{\pi_n^{10}(t_n)}\right\}. \tag{A5}$$

To summarize, we have now obtained a linear program, where we want to maximize the objective function in (A3) with respect to the two variables q_n^{00} and q_n^{10}, subject to the constraints (A4) and (A5).

If for some fixed $t_n \in [t_n^{\min}, t_n^{\max}]$ we consider a coordinate system with q_n^{00} along the first axis and q_n^{10} along the second axis, the constraints in (A4) and (A5) form a rectangular region of feasible solutions, with two edges in the q_n^{00}-direction and two edges in the q_n^{10}-direction. The optimal solution lies in a corner point of this region. Since $\pi_n^{00}(t_n)$ and $\pi_n^{10}(t_n)$ are nonnegative for any $t_n \in [t_n^{\min}, t_n^{\max}]$, it is easily seen from (A3) that $E_n(t_n, q_n)$ is maximized with respect to q_n when q_n^{00} and q_n^{10} are as large as possible. Consequently, the optimal solutions of q_n^{00} and q_n^{10} must equal the upper bounds in (A4) and (A5), corresponding to the upper right corner of the rectangular feasible region. That is,

$$q_n^{*00}(t_n) = \min\left\{1, \frac{f_n^{00}}{\pi_n^{00}(t_n)}\right\},$$
$$q_n^{*10}(t_n) = \min\left\{1, \frac{f_n^{10}}{\pi_n^{10}(t_n)}\right\}.$$

Clearly, $q_n^{*00}(t_n)$ and $q_n^{*10}(t_n)$ are continuous and piecewise-defined functions of t_n, since $\pi_n^{00}(t_n)$ and $\pi_n^{10}(t_n)$ are linear functions of t_n. Specifically, for t_n-values such that $\pi_n^{00}(t_n) > f_n^{00}$, we get $q_n^{*00}(t_n) = f_n^{00}/\pi_n^{00}(t_n)$, while for t_n-values such that $\pi_n^{00}(t_n) \le f_n^{00}$, we get $q_n^{*00}(t_n) = 1$. Likewise, for t_n-values such that $\pi_n^{10}(t_n) > f_n^{10}$, we get $q_n^{*10}(t_n) = f_n^{10}/\pi_n^{10}(t_n)$, while for t_n-values such that $\pi_n^{10}(t_n) \le f_n^{10}$, we get $q_n^{*10}(t_n) = 1$.

Inserting the optimal solutions $q_n^{*00}(t_n)$ and $q_n^{*10}(t_n)$ into (A3), returns $E_n^*(t_n)$. Doing this, it is easily seen that $E_n^*(t_n)$ is a CPL function of t_n, consisting of maximally three pieces, each piece having a slope equal to either −2, 0, or 2.

The intermediate iterations

At each intermediate iteration of the backward recursion, we are dealing with a parametric, *piecewise* linear program, whose analytic solution is, generally, more intricate than that of the parametric linear program of the first iteration. However, proving that the resulting function $E_{k:n}^*(t_k)$ is CPL, provided that $E_{k+1:n}^*(t_{k+1})$ is CPL, is not too complicated. Below, we present a proof which can be summarized as follows. First, for each subproblem $j \in \mathcal{S}_{k+1}$ corresponding to the jth linear piece of the previous CPL function $E_{k+1:n}^*(t_{k+1})$, we explain that the corners (or possibly edges) of the feasible region that may represent the optimal solution yield a CPL function in t_k when inserted into the objective function $E_{k:n}^{(j)}(t_k, q_k)$. Second, we argue that since the boundary of the feasible region evolves in a continuous way as a function of t_k and since also $E_{k:n}^{(j)}(t_k, q_k)$ is continuous in t_k and q_k, any infinitesimal change in t_k can only induce an infinitesimal change in the location of the optimal solution. Third, we conclude from these observations that $\tilde{E}^{(j)}_{k:n}(t_k)$ is CPL for each subproblem $j \in \mathcal{S}_{k+1}$. This means that the final function $E^*_{k:n}(t_k)$ is the maximum of multiple CPL functions. Therefore, $E^*_{k:n}(t_k)$ itself must be piecewise linear. The additional fact that $E^*_{k:n}(t_k)$ is continuous is an immediate consequence of the continuity of the whole optimization problem and the connection between the subproblems.

As in the first backward step, the equality constraints (31) for q_k allow us to reformulate the optimization problem in terms of the two variables q^{00}_k and q^{10}_k. Specifically, for each subproblem $j \in \mathcal{S}_{k+1}$, we can use the equality constraints to write the objective function $E^{(j)}_{k:n}(t_k, q_k)$ cf. (35) in terms of q^{00}_k and q^{10}_k as

$$E^{(j)}_{k:n}(t_k, q_k) = \tilde{\beta}^{(j)}_k \pi^{00}_k(t_k) q^{00}_k + \tilde{\beta}^{(j)}_k \pi^{10}_k(t_k) q^{10}_k + \tilde{\alpha}^{(j)}_k, \tag{A6}$$

where

$$\tilde{\beta}^{(j)}_k = 2 + b^{(j)}_{k+1}(\rho^{0|0}_k - \rho^{0|1}_k)$$

and

$$\tilde{\alpha}^{(j)}_k = f(x_k = 1) - f(x_k = 0|y) + a^{(j)}_{k+1} + b^{(j)}_{k+1}(f^{00}_k + f^{10}_k)\rho^{0|1}_k.$$

The corresponding constraints for q^{00}_k and q^{10}_k read

$$\max\left\{0, \frac{f^{00}_k - \pi^{01}_k(t_k)}{\pi^{00}_k(t_k)}\right\} \leq q^{00}_k \leq \min\left\{1, \frac{f^{00}_k}{\pi^{00}_k(t_k)}\right\}, \tag{A7}$$

$$\max\left\{0, \frac{f^{10}_k - \pi^{11}_k(t_k)}{\pi^{10}_k(t_k)}\right\} \leq q^{10}_k \leq \min\left\{1, \frac{f^{10}_k}{\pi^{10}_k(t_k)}\right\}, \tag{A8}$$

and

$$t^{B(j)}_{k+1} \leq (\rho^{0|0}_k - \rho^{0|1}_k)\pi^{00}_k(t_k)q^{00}_k + (\rho^{0|0}_k - \rho^{0|1}_k)\pi^{10}_k(t_k)q^{10}_k + (f^{00}_k + f^{10}_k)\rho^{0|1}_k \leq t^{B(j+1)}_{k+1}. \tag{A9}$$

If for some fixed $t_k \in [t^{\min}_k, t^{\max}_k]$ we consider a coordinate system with q^{00}_k along the first axis and q^{10}_k along the second axis, we see that the feasible region formed by the constraints (A7)–(A9) is a polygon with maximally six corners. The region is enclosed by two lines in the q^{00}_k-direction cf. (A7), two lines in the q^{10}_k-direction cf. (A8), and two parallel lines with a negative slope of $-\pi^{00}_k(t_k)/\pi^{10}_k(t_k)$ cf. (A9). Figure 8 illustrates some of the possible shapes that the region can take. Clearly, the optimal solution is located in a corner of the feasible region, possibly along a whole edge.

To understand where along the boundary of the feasible region the optimal solution is located, we note from (A6) that if $\tilde{\beta}^{(j)}_k$ is positive, then $E^{(j)}_{k:n}(t_k, q_k)$ is maximized when q^{00}_k and q^{10}_k are as large as possible, while if $\tilde{\beta}^{(j)}_k$ is negative, then $E^{(j)}_{k:n}(t_k, q_k)$ is maximized when q^{00}_k and q^{10}_k are as small as possible. For simplicity, we assume in the following that the feasible region is nonempty.

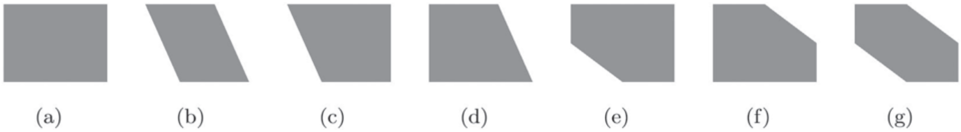


FIGURE 8 Illustrations of some possible shapes for the feasible regions of the linear programs at the intermediate steps of the backward recursion. The polygons are drawn in a coordinate system with q_k^{00} in the horizontal direction and q_k^{10} in the vertical direction

First, consider the case with $\tilde{\beta}_k^{(j)}$ positive. Then, we need to check whether or not the upper of the two lines corresponding to the two inequality constraints in (A9) forms an edge of the feasible region. If this line does *not* form an edge of the feasible region, see, for example, the shapes in Figure 8a,c,e; we observe that the point $(q_k^{00(\mathcal{U})}(t_k), q_k^{10(\mathcal{U})}(t_k))$, where

$$q_k^{00(\mathcal{U})}(t_k) = \min\left\{1, \frac{f_k^{00}}{\pi_k^{00}(t_k)}\right\}, \tag{A10}$$

$$q_k^{10(\mathcal{U})}(t_k) = \min\left\{1, \frac{f_k^{10}}{\pi_k^{10}(t_k)}\right\}, \tag{A11}$$

is a corner. Moreover, this corner represents the optimal solution, since q_k^{00} and q_k^{10} jointly take their maximal values in this point. Now, if we insert the functions in (A10) and (A11) into the objective function $E_{k:n}^{(j)}(t_k, q_k)$, we obtain a CPL function in t_k. Thereby, given that (A10) and (A11) represent a corner of the feasible region for all values of t_k, the resulting function $\tilde{E}_{k:n}^{(j)}(t_k)$ is CPL in t_k. If, on the other hand, the upper of the two lines of the constraints (A9) *does* represent an edge of the feasible region, see for instance Figure 8b,d,f, g; then this whole edge represents the optimal solution. That is, any point along the edge is optimal. This result is due to that the slope of the objective function and the slope of the line for this edge are equal, from which it follows that the objective function takes the same maximal value anywhere along the edge. Now, if we insert (q_k^{00}, q_k^{10})-coordinates located on the edge into the objective function $E_{k:n}^{(j)}(t_k, q_k)$, we get a function which is constant, and hence CPL, in t_k. Thereby, given that the edge is part of the feasible region for all values of t_k, the resulting function $\tilde{E}_{k:n}^{(j)}(t_k)$ is CPL in t_k. Next, consider the case with $\tilde{\beta}_k^{(j)}$ negative. Then, the situation is equivalent to the case with $\tilde{\beta}_k^{(j)}$ positive, but we need to consider the lower part of the feasible region instead of the upper. That is, we need to check whether or not the lower of the two lines corresponding to the constraints in (A9) forms an edge of the feasible region. If this line does *not* represent an edge, see, for example, Figure 8a,d,f; the optimal solution is found in the lower left corner point, $(q_k^{00(\mathcal{L})}(t_k), q_k^{10(\mathcal{L})}(t_k))$, where

$$q_k^{00(\mathcal{L})}(t_k) = \max\left\{0, \frac{f_k^{00} - \pi_k^{01}(t_k)}{\pi_k^{00}(t_k)}\right\}, \tag{A12}$$

$$q_k^{10(\mathcal{L})}(t_k) = \max\left\{0, \frac{f_k^{10} - \pi_k^{11}(t_k)}{\pi_k^{10}(t_k)}\right\}. \tag{A13}$$

Again, if we insert the functions in (A12) and (A13) into the objective function $E^{(j)}_{k:n}(t_k, q_k)$, we obtain a CPL function in t_k. Thereby, given that (A12) and (A13) represent a corner of the feasible region for all values of t_k, the resulting function $\tilde{E}^{(j)}_{k:n}(t_k)$ is CPL in t_k. If, on the other hand, the lower of the two lines of the constraints (A9) *does* represent an edge of the feasible region, then this edge also represents the optimal solution since the objective function takes the same maximal value anywhere along this edge. Now, if we insert (q^{00}_k, q^{10}_k)-coordinates located on the optimal edge into the objective function $E^{(j)}_{k:n}(t_k, q_k)$, we obtain a function which is constant, and hence CPL, in t_k. Thereby, given that the edge is part of the feasible region for all values of t_k, the resulting function $\tilde{E}^{(j)}_{k:n}(t_k)$ is CPL in t_k.

Because the objective function, $E^{(j)}_{k:n}(t_k, q_k)$, as well as all the constraints (A7)–(A9) are continuous in t_k and q_k, it follows that any infinitesimal change δt_k in t_k can only induce corresponding infinitesimal changes in the shape of the feasible region and the value of the objective function. Hence, the optimal solution at any t_k-value t'_k must be located in the same corner (or along the same edge) as the optimal solution at the t_k-value $t'_k + \delta t_k$. We note, however, that it is possible that the infinitesimal change δt_k may have added or deleted an edge from the region. In this case, it is possible that a single corner represented the optimal solution at t'_k, while a whole edge represents the optimal solution at $t'_k + \delta t_k$, or vice versa. However, this will not cause any discontinuities in the resulting function $\tilde{E}^{(j)}_{k:n}(t_k)$ because of the continuity of the optimization problem as a whole. We have already showed that the coordinates describing the evolution of every potentially optimal corner (or edge) as a function of t_k return a CPL function in t_k. Hence, we understand that $\tilde{E}^{(j)}_{k:n}(t_k)$ must be CPL.

Finally, we obtain the function $E^*_{k:n}(t_k)$ by taking the maximum of the $\tilde{E}^{(j)}_{k:n}(t_k)$'s. Taking the maximum of a set of continuous piecewise linear functions necessarily produces another piecewise linear, but not necessarily a continuous, function. However, it is obvious without a further proof that $E^*_{k:n}(t_k)$ must be continuous, since all functions in the whole optimization problem are continuous. Thereby, we can conclude that $E^*_{k:n}(t_k)$ is CPL.

According to numerical experiments, it seems that $q^{*00}_k(t_k)$ and $q^{*10}_k(t_k)$ are analytically given as $q^{*00}_k(t_k) = q^{00(\mathcal{U})}_k(t_k)$ and $q^{*10}_k(t_k) = q^{10(\mathcal{U})}_k(t_k)$, just as in the first backward iteration. However, we have not proved this result, since it is not really important for our application. Yet, we note that if this result can be proved, the computation of $q(\tilde{x}|x, y)$ becomes particularly simple.

Computing the breakpoints of $E^*_{k:n}(t_k)$

This section concerns computation of the breakpoints of the CPL function $E^*_{k:n}(t_k)$ at each intermediate iteration $2 \leq k < n$ of the backward recursion. The breakpoints of $E^*_{k:n}(t_k)$ should be computed prior to solving the corresponding parametric piecewise linear program in order to avoid numerical computation of $E^*_{k:n}(t_k)$ on a grid of t_k-values. However, it can in some cases be a bit cumbersome and technical to compute the explicit set of t_k-values representing the breakpoints of $E^*_{k:n}(t_k)$. Fortunately, it is an easier task to compute a slightly larger set of t_k-values representing *potential* breakpoints of $E^*_{k:n}(t_k)$, which includes all of the *actual* breakpoints. For convenience, we denote in the following the set of actual breakpoints by A_k and the larger set of potential breakpoints by $A'_k \supset A_k$. Having computed the set A'_k, we can solve our parametric piecewise linear program for the t_k-values in this set, and afterward go through the values of the resulting function $E^*_{k:n}(t_k)$ to check which of the elements in A'_k that represent *actual* breakpoints that must be stored in A_k, and which points that can be omitted.

As explained in Section A1, the function $E^*_n(t_n)$ of the first backward iteration consists of maximally three linear pieces. Hence it has maximally two breakpoints in addition to its two endpoints

$t_n^{\min}$ and $t_n^{\max}$. Since at each intermediate iteration we consider a more complicated parametric *piecewise* linear program, additional breakpoints can occur in $E^*_{k:n}(t_k)$, with the number of possible breakpoints for $E^*_{k:n}(t_k)$ increasing with the number of breakpoints for $E^*_{k+1:n}(t_k)$ computed at the previous step of the recursion. To compute the set A'_k of potential breakpoints for $E^*_{k:n}(t_k)$, we need to check for which t_k-values the corners of the rectangular region formed by the constraints in (A7) and (A8) intersect with the lines of the constraints in (A9) for each $j \in \mathcal{S}_{k+1}$. Each t_k-value that causes such an intersection must be included in the set A'_k. To understand why, consider a subproblem $j \in \mathcal{S}_{k+1}$, and assume $\tilde{\beta}_k^{(j)}$ is positive. Furthermore, suppose that for all $t_k \in [t_k^{\min}, t_k^{\max}]$ the feasible region has a rectangular shape as shown in Figure 8a, meaning that the region is only enclosed by the constraints (A7) and (A8), while the extra constraints in (A9) do not contribute to the shape of the region. Then, from Section A2, we know that the optimal solution lies in the upper right corner given by (A10) and (A11) for all t_k. Moreover, we know that $\tilde{E}^{(j)}_{k:n}(t_k)$ is CPL with breakpoints corresponding to the breakpoints of (A10) and (A11). Now, suppose instead that after some specific value t'_k the shape of the feasible region changes from a rectangular shape as in Figure 8a to a pentagon shape as in Figure 8f. This means that the upper of the two lines formed by the extra constraints in (A9) at the t_k-value t'_k intersects with the upper right corner point given by (A10) and (A11), while for $t_k > t'_k$ the constraints results in that an extra edge is added to the feasible region. From Section A2, we then know that for $t_k > t'_k$ this extra edge represents the optimal solution and the value of the objective function remains constant as a function of $t_k > t'_k$. Thereby, we understand that a breakpoint may occur in $\tilde{E}^{(j)}_{k:n}(t_k)$, and hence possibly in $E^*_{k:n}(t_k)$, at the t_k-value t'_k. If the feasible region were to evolve in a different way than the one considered here, similar arguments can be formulated. In A'_k, we must also include the breakpoints of the functions in (A10)–(A13), that is, the breakpoints of the functions describing the coordinates for the lower left and upper right corner points of the feasible region when the constraints (A9) do not contribute.

Paper II

Geophysics-based fluid-facies predictions using ensemble updating of binary state vectors

Margrethe Kvale Loe, Dario Grana and Håkon Tjelmeland

Published in *Mathematical Geosciences*

Math Geosci (2021) 53:325–347
https://doi.org/10.1007/s11004-021-09922-4

SPECIAL ISSUE

Geophysics-Based Fluid-Facies Predictions Using Ensemble Updating of Binary State Vectors

Margrethe Kvale Loe[1] · **Dario Grana**[2] · **Håkon Tjelmeland**[1]

Received: 22 January 2020 / Accepted: 12 January 2021 / Published online: 8 February 2021

Abstract Fluid flow simulations are commonly used to predict the fluid displacement in subsurface reservoirs; however, model validation is challenging due to the lack of direct measurements. Geophysical data can be used to monitor the displacement of the fluid front. The updating of the fluid front location in two-phase flow problems based on time-lapse geophysical data can be formulated as an inverse problem, specifically a data assimilation problem, where the state is a vector of binary variables representing the fluid-facies and the observations are measurements of continuous geophysical properties, such as electrical or elastic properties. In geosciences, many data assimilation problems are solved using ensemble-based methods relying on the Kalman filter approach. However, for discrete variables, such approaches cannot be applied due to the Gaussian-linear assumption. An innovative approach for mixed discrete-continuous problems based on ensemble updating of binary state vectors is presented for fluid-facies prediction problems with time-lapse geophysical properties. The proposed inversion method is demonstrated in a synthetic two-dimensional simulation example where water is injected into a reservoir and hydrocarbon is produced. Resistivity values obtained from controlled-source electromagnetic data are assumed to be available at different times. According to the results, the proposed inversion method is to a large extent able to reproduce the true underlying binary field of fluid-facies.

Keywords Data assimilation · Ensemble methods · Fluid-facies prediction · Markov chain models

Margrethe Kvale Loe
margrethe.loe@ntnu.no

[1] Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway

[2] Department of Geology and Geophysics, University of Wyoming, Laramie, WY, USA

1 Introduction

Monitoring the spatial distribution of fluids in the subsurface is a challenging modeling problem due to the uncertainty in rock and fluid properties. If the petrophysical properties, (e.g., porosity and permeability) and the subsurface conditions (e.g., temperature and pressure) are known, the fluid volumes and their spatial distribution can be predicted by solving partial differential equations for fluid flow in porous media (Bear 2013). However, in practical applications, the majority of the model parameters cannot be measured directly and can only be approximated from indirect geophysical measurements (Aki and Richards 1980; Doyen 2007). Hence, the accuracy in the model predictions depends on the quality of the data and the approximations in the physical models. For these reasons, it is necessary to update the fluid models every time new geophysical data are available.

One of the main challenges in monitoring fluid flow in the subsurface is obtaining accurate and precise predictions of the fluid front at different times. Geophysical surveys, acquired at the surface, provide measurements of physical properties whose changes in time reflect variations in rock and fluid properties. Hence, the location of the fluid front can be updated based on time-lapse geophysical data and fluid flow simulations. Examples of time-lapse geophysical surveys include seismic data that depend on changes in elastic properties and electromagnetic data that depend on changes in electrical properties.

Changes in fluid saturations modify the elastic and electrical response of reservoir rocks. Therefore, unknown fluid flow saturations can be predicted as the solution of an inverse problem where the data are geophysical observations and the governing equations are geophysical models (Doyen et al. 2000). However, the resolution of geophysical data measured at the surface is generally lower than the desired resolution of the fluid distribution model due to the bandlimited nature of the signal and the relatively low signal-to-noise ratio. Therefore, prediction of the fluid saturations is highly uncertain and saturation models can be inaccurate. In many applications, geophysical data can be used to interpret the fluid phase at a specific location; however, it is generally difficult to accurately predict fluid percentages. For this reason, rather than using continuous variables representing fluid volumes or fluid saturations, in this work, the inverse problem is formulated in terms of a discrete variable. For multiphase flow problems, the discrete variable represents fluid-facies (or fluid units), such as water-, air-, hydrocarbon-, or CO_2-saturated rocks, depending on the application. In particular, for two-phase flow problems, fluid-facies can be represented by a binary variable. The proposed methodology is defined in a probabilistic setting; therefore, the model property of interest is a discrete random variable.

In this work, the focus is on fluid properties that cause changes in the electrical response measured by electromagnetic data, such as electrical resistivity tomography (ERT) and controlled-source electromagnetic (CSEM). The goal of this work is to predict fluid-facies in subsurface reservoirs based on estimated resistivity values from time-lapse geophysical data. From a mathematical point of view, this is a data assimilation problem where the unknown state is a discrete random variable (the fluid-facies) and the observed data are geophysical measurements (the resistivity values). The resistivity of fluid-saturated porous rocks is obtained from electromagnetic measurements.

Here, it is assumed that resistivity values have been computed from CSEM data as a result of a preliminary inversion process. In near surface geophysics applications, ERT data are commonly acquired to investigate the seasonal water displacement (Flinchum et al. 2018; Kotikian et al. 2019; Claes et al. 2020). In hydrocarbon exploration and carbon sequestration, CSEM data are often measured to study the fluid spatial distribution and monitor the fluid front location (Weitemeyer et al. 2006; Lien and Mannseth 2008; Orange et al. 2009; Constable 2010; MacGregor 2012; Lien et al. 2014). The prediction of resistivity from electromagnetic data is itself an inverse problem (Gasperikova and Hoversten 2006; Buland and Kolbjørnsen 2012; MacGregor 2012; Bhuyian et al. 2012). Probabilistic and deterministic approaches have been proposed with different model parameterizations, in terms of resistivity or saturation. Time-lapse electrical resistivity inversions have been proposed in several applications (Berre et al. 2011; Shahin et al. 2012; Lien et al. 2014; Tveit et al. 2015; Commer et al. 2016; Bergmann et al. 2017).

Classification methods for litho-facies and fluid-facies based on geophysical data have also been proposed. Several clustering analysis methods described in Hastie et al. (2009) have been used for facies classification in geophysics inverse problems (Doyen 2007). Classification methods include supervised and unsupervised techniques (Hastie et al. 2009; Martinez and Martinez 2015). Clustering and pattern recognition methods have been used to classify geophysical measurements. However, the majority of these applications focuses on static characterization problems, with a spatial correlation component (i.e. facies are spatially correlated to mimic the geological continuity) but without a temporal component (i.e. facies are predicted at a given time step, typically before dynamic processes start).

The focus of this work is time-dependent fluid-facies characterization problems where the spatial distribution of fluid-facies changes in time and is monitored using time-lapse geophysical properties. Therefore, in this study a dynamic fluid-facies classification is presented, and it is applicable to geological dynamic problems where one fluid (e.g., water) replaces another fluid (e.g., hydrocarbon) in rock formations. Reservoir modeling with time-lapse data is a data assimilation problem where the model variables are predicted and updated when new measurements become available. Several stochastic optimization methods have been proposed for data assimilation problems, and during the last decade, ensemble-based methods have become the most popular stochastic data assimilation method in geoscience applications.

Data assimilation can refer to a range of different inference procedures, of which the two most common are filtering and smoothing. In the present article, the focus is exclusively on the filtering problem. There are two main classes of ensemble-based filtering methods: particle filters (Doucet et al. 2001) and ensemble Kalman filters (EnKFs) (Evensen 2009). Hybrid versions of these filters have also been proposed. Particle filters have the advantage of being exact in the sense that as the ensemble size goes to infinity, the ensemble representations of the series of filtering distributions converge to the corresponding correct series of distributions. Particle filters are also very general as they do not rely on any specific assumptions about the distributions of the model variables. Hence, particle filter methods are, in principle, applicable to both discrete and continuous variables. In practice, however, particle filters are known to collapse when the dimension of the state vector is large. The EnKF is a filtering method which relies on a linear-Gaussian assumption about the underlying model. Despite the linear-Gaussian assumption, studies show that the EnKF provides good results even in non-linear, non-Gaussian problems, and unlike the particle filter it also scales well to problems with very high-dimensional state vectors. The EnKF has been applied to geophysical data assimilation and history matching problems using seismic and electromagnetic data (e.g. Tveit et al. 2015; Tveit et al. 2020). Recent publications focus on the integration of fluid flow simulation and geophysical data assimilation for the monitoring of the fluid front location (Trani et al. 2012; Leeuwenburgh and Arts 2014; Zhang and Leeuwenburgh 2017). However, the Gaussian approximations make the EnKF applicable only in situations with continuous variables. For problems with discrete variables, such as fluid-facies classification, the filter is not appropriate.

Ensemble filtering of discrete variables is a challenging problem which has received fairly little attention in the literature compared to filtering of continuous variables. Oliver et al. (2008) propose a strategy where the EnKF is used to update the discrete variables. Specifically, they propose a two-step strategy where, in the first step, the EnKF is used to update the discrete variables as if they were continuous, and in the second step, the updated continuous-valued variables are mapped back to the original discrete state space using the Viterbi algorithm (Viterbi 1967). Loe and Tjelmeland (2020) present an alternative updating method for binary vectors in one-dimensional space based on a generalized approach of the EnKF. Instead of using a linear-Gaussian model assumption in the ensemble update, as in the EnKF, they construct an update based on a hidden Markov model assumption. To capture as much information as possible from the forecast ensemble, including potential non-Markov properties, the expected number of components of the binary state vector that remain unchanged is maximized.

This paper presents an ensemble-based data assimilation method for a problem where the state vector at each time step is a vector of binary variables and the observations are continuous-valued estimated resistivity values. The binary variables of the state vector represent two different fluid-facies, for example water and hydrocarbon or CO_2, and each binary variable is connected to a continuous-valued variable representing water saturation. High water saturation values indicate the presence of the water facies, while low saturation values indicate the presence of the other fluid-facies. The proposed ensemble filtering method alternates between a forecast step performed in the continuous state space of the saturation variable and an update step performed in the discrete state space of the fluid-facies variable, and between each step an appropriate mapping from one state space to the other is performed. The update step is performed according to the updating procedure for binary state vectors proposed in Loe and Tjelmeland (2020). The proposed inversion method is demonstrated in a synthetic two-dimensional example representing a two-phase flow problem with resistivity values available at different times. According to the results, the proposed procedure is to a large extent able to reproduce the true underlying binary field of fluid-facies. Larger ensemble sizes provide more accurate results, but the results obtained with smaller ensemble sizes are also satisfactory.

The remains of this paper takes the following outline. First, Sect. 2 formulates the inverse problem more formally and presents the proposed ensemble-based inversion method. Next, Sect. 3 presents numerical results based on a two-dimensional synthetic model for a two-phase fluid flow problem. Finally, a few closing remarks are given in Sect. 4.

2 Methodology

2.1 Inverse Problem Setting

The problem addressed in this work is the prediction of fluid-facies from time-lapse resistivity values. Consider a time series $\{k_i\}_{i=1}^{N_t}$ where $k_i = (k_i^1, \dots, k_i^{N_k})$ represents an N_k-dimensional vector of fluid-facies at a certain time t_i, $i = 1, \dots, N_t$ in a reservoir. Each component k_i^j of k_i can take a value in the set $\{0, 1, \dots, K-1\}$, where K is the number of different fluid-facies. Given a corresponding series of resistivity data $\{d_i\}_{i \in \mathbb{T}}$, where $d_i = (d_i^1, \dots, d_i^{N_d}) \in \mathbb{R}^{N_d}$ is an N_d-dimensional vector of resistivity measurements recorded at time t_i, and $\mathbb{T} \subseteq \{1, \dots, N_t\}$, the goal is, for each time step $i = 1, \dots, N_t$, to assess the distribution of fluid-facies k_i in the reservoir. Notice from the set $\mathbb{T}$ that an observation d_i may be available at every time step $i = 1, \dots, N_t$, or just a subset of them.

In this work, each component k_i^j of k_i is assumed to be connected to a continuous variable $m_i^j \in [s_{wi}, 1]$ representing water saturation, where s_{wi} is an irreducible water saturation value; that is, the fraction of water that a porous rock can retain due to non-connected porosity, low permeability and/or capillary forces. Here, the irreducible water saturation value $s_{wi} = 0.2$ is assumed. Given $\{m_i\}_{i=1}^{N_t}$, the resistivity data $\{d_i\}_{i \in \mathbb{T}}$ are assumed to be conditionally independent, so that the vector d_i at time t_i depends only on m_i according to

$$d_i = f(m_i, e_i), \tag{1}$$

where f is a known, possibly non-linear function, and the variable $e_i \in \mathbb{R}^{N_e}$ is an N_e-dimensional vector of measurement random errors assumed to follow a known probability distribution. Similarly, the saturation m_{i+1} at time step $i+1$, given all the saturation values $m_1, \dots m_i$ up to time step i, depends only on m_i according to a known forward model,

$$m_{i+1} = g(m_i), \tag{2}$$

for $i = 1, \dots, N_t$, where g is the fluid flow simulation, generally given by a system of partial differential equations solved by finite difference methods (Aziz 1979).

The goal of this work is, for each time step $i = 1, \dots, N_t$, to assess the filtering distribution $p(k_i | d_{1:i})$, where $d_{1:i} = \left\{d_j;\, j \in \mathbb{T} \cap \{j \leq i\}\right\}$, that is, the distribution of fluid-facies k_i given all the resistivity data up to time t_i. Only $K = 2$ fluid-facies are assumed in this work: facies 1 represents water and facies 0 represents another fluid-facies. The relationship between the fluid-facies $k_i^j \in \{0, 1\}$ and the saturation value $m_i^j \in [s_{wi}, 1]$ is assumed as

$$k_i^j = \begin{cases} 0, & \text{if } m_i^j \in [s_{wi}, r], \\ 1, & \text{if } m_i^j \in (r, 1], \end{cases} \tag{3}$$

where $r \in (s_{wi}, 1)$ is some appropriate threshold. The value of the parameter r might vary from one application to another. A reasonable choice is to set $r = 0.5$ such that each fluid facies is named after the predominant fluid component. However, from a reservoir management perspective, the focus is generally on areas with a high concentration of hydrocarbon. Therefore one could choose a lower value to identify the regions that are economically valuable.

2.2 Forward Model

The prediction of the time-dependent electrical response of a reservoir model requires a rock-physics model to link the petrophysical properties, such as porosity and fluid saturations, to the resistivity of the saturated porous rocks and a fluid flow simulation model to compute the saturation at a given time step, given the saturation at the previous time step. In the proposed approach, porosity and permeability are assumed to be estimated from pre-injection geophysical measurements (e.g., seismic data). Alternatively, multiple geostatistical simulations of porosity and permeability can be generated to repeatedly apply the methodology to an ensemble of realizations; however, the computational cost would linearly increase with the number of realizations.

A rock-physics model is a relationship to predict the geophysical response of saturated porous rocks. Assuming that the porosity ϕ of the porous rock is known, the resistivity R (the measured data d in the inverse problem) of the porous rock saturated with water saturation s_w can be predicted using Archie's law (Mavko et al. 2009),

$$R = \frac{R_w}{\phi^a s_w^b}, \tag{4}$$

where R_w is the resistivity of formation water, a is the cementation exponent, and b is the saturation exponent (Mavko et al. 2009). The parameters R_w, a and b in Eq. (4) are assumed to be constant in time. Archie's equation is valid for clean sandstone formations. For formations with a small to medium clay volume, Archie's equation can be modified to account for the conductivity of the clay mineral as in Simandoux and Poupon-Leveaux models (Mavko et al. 2009).

The dynamic model that governs two-phase fluid flow in porous media is based on the constitutive equations of mass and momentum balance. The model is numerically solved using the black-oil framework to predict the saturation and pressure at each time step, given the initial rock and fluid parameters (Aziz 1979). In this work, the MATLAB Reservoir Simulation Toolbox (Lie 2019) is adopted.

2.3 Inversion Method

To solve the inverse problem presented above, an ensemble-based strategy where the forecast step is performed in the continuous domain of m_i and the update step is performed in the discrete domain of k_i is adopted. At each time t_i, an ensemble of fluid-facies fields $\left\{k_i^{(1)}, \ldots, k_i^{(M)}\right\}$ represents the distribution of k_i given the resistivity data up to time t_{i-1}, that is $d_{1:i-1}$. Likewise an ensemble of saturation fields $\left\{m_i^{(1)}, \ldots, m_i^{(M)}\right\}$ represents the distribution of m_i given the same resistivity data. Correspondingly, the distributions of k_i and m_i given resistivity data up to time t_i, that is $d_{1:i}$, are also represented by ensembles, which are denoted by $\left\{\tilde{k}_i^{(1)}, \ldots, \tilde{k}_i^{(M)}\right\}$ and $\left\{\tilde{m}_i^{(1)}, \ldots, \tilde{m}_i^{(M)}\right\}$, respectively. The main steps of the inversion procedure are summarized in Algorithm 1, while each step is studied in closer detail in the following sections.

```
Initialize: $\left\{m_1^{(1)}, \ldots, m_1^{(M)}\right\}$.
For $i = 1, \ldots, N_t$ do
1. Update:
    (a) Map $m_i^{(l)} \to k_i^{(l)}, l = 1, \ldots, M$, using Eq. (3).
    (b) Update $k_i^{(l)} \to \tilde{k}_i^{(l)}, l = 1, \ldots, M$, as discussed in Sect. 2.3.1.
2. Forecast:
    (a) Map $\tilde{k}_i^{(l)} \to \tilde{m}_i^{(l)}, l = 1, \ldots, M$, as discussed in Sect. 2.3.2.
    (b) Generate $m_{i+1}^{(l)} = g(\tilde{m}_i^{(l)}), l = 1, \ldots, M$, using Eq. (2).
End
```

Algorithm 1: Inversion method

2.3.1 The Update Step

As summarized in Algorithm 1, the update step of the proposed approach involves two parts. First, the ensemble $\left\{m_i^{(1)}, \ldots, m_i^{(M)}\right\}$ is mapped to a corresponding ensemble $\left\{k_i^{(1)}, \ldots, k_i^{(M)}\right\}$ using the assumed relation between k_i and m_i in Eq. (3). Second, $\left\{k_i^{(1)}, \ldots, k_i^{(M)}\right\}$ is updated to take the new observation d_i at time t_i into account. In the following, the two parts of the update step are discussed in more detail.

The ensemble of saturation fields $\left\{m_i^{(1)}, \ldots, m_i^{(M)}\right\}$ is mapped to a corresponding ensemble of fluid-facies fields $\left\{k_i^{(1)}, \ldots, k_i^{(M)}\right\}$ by simply applying Eq. (3) to each element in each of the ensemble members, that is, set

$$k_i^{(l),j} = \begin{cases} 0, & \text{if } m_i^{(l),j} \in [s_{wi}, r], \\ 1, & \text{if } m_i^{(l),j} \in (r, 1], \end{cases} \tag{5}$$

for each location $j = 1, \ldots, N_k$ for each ensemble member $l = 1, \ldots, M$.

To update the ensemble $\left\{k_i^{(1)}, \ldots, k_i^{(M)}\right\}$ to take the new observation d_i into account, the procedure proposed in Loe and Tjelmeland (2020) is adapted to the situation considered in the present article. In the present article, it is assumed that the fluid-facies k_i at each time step $i = 1, \ldots, N_t$ is defined on a two-dimensional lattice. However, the method in Loe and Tjelmeland (2020) is applicable only for vectors with a one-dimensional spatial arrangement. Therefore, in order to apply their procedure, the updating of each column in the lattice is done independently of the others. Of course, this is not an ideal approach since it means that some of the spatial correlation in the horizontal direction is lost; however, since the forecast step incorporates spatial correlation in both directions, one may still obtain satisfactory results. Let C denote the number of columns in the lattice and let $k_{i,c}^{(l)}$ and $\tilde{k}_{i,c}^{(l)}$ for $c = 1, \ldots, C$ denote the values in column number c of $k_i^{(l)}$ and $\tilde{k}_i^{(l)}$, respectively. The procedure used for the updating of $k_{i,c}^{(l)}, l = 1, \ldots, M$ is inspired by the updating procedure used in the ensemble Kalman filter (EnKF), but as the elements of $k_{i,c}^{(l)}$ are binary variables, the updating procedure is based on a first order Markov chain instead of a Gaussian distribution as in EnKF. Thus, the update of $k_{i,c}^{(l)}, l = 1, \ldots, M$ starts by estimating a (non-stationary) Markov chain for column c. Using a Bayesian model for this estimation, the $k_{i,c}^{(l)}, l = 1, \ldots, M$ are considered as independent realizations from the assumed Markov chain, and independent uniform priors on the unit interval are adopted for the initial distribution and for each transition probability. The maximum a posteriori estimators are then used to estimate the initial distribution and the transition probabilities. The estimated Markov chain is used as a prior distribution in a new Bayesian model and combined with an assumed likelihood model for the part of d_i related to column c. It is here assumed that d_i contains one component d_i^j for each element k_i^j in k_i, that the components of d_i are conditionally independent given k_i, and that d_i^j depends only on k_i through k_i^j. The likelihood for the part of d_i related to column c can then be expressed as

$$p(d_{i,c}|k_{i,c}) = \prod_{j \text{ in column } c} p(d_i^j|k_i^j). \tag{6}$$

The likelihood model $p(d_i^j|k_i^j)$ is specified by first defining d_i^j to be given by replacing the saturation value s_w in the rock physics model in Eq. (4) by an auxiliary random variable $u_i^j \in [s_{wi}, 1]$, that is,

$$d_i^j = \frac{R_w}{\phi^a \left(u_i^j\right)^b}. \tag{7}$$

The distribution of the latent u_i^j should depend on the fluid-facies value k_i^j, and it is assumed that

$$p_0(u_i^j) = p(u_i^j | k_i^j = 0) = \begin{cases} c_0 e^{-\lambda_0 r} & \text{when } u_i^j \in [s_{wi}, r], \\ c_0 e^{-\lambda_0 u_i^j} & \text{when } u_i^j \in (r, 1], \end{cases} \tag{8}$$

and

$$p_1(u_i^j) = p(u_i^j | k_i^j = 1) = \begin{cases} c_1 e^{\lambda_1 u_i^j} & \text{when } u_i^j \in [s_{wi}, r], \\ c_1 e^{\lambda_1 r} & \text{when } u_i^j \in (r, 1], \end{cases} \tag{9}$$

where c_0 and c_1 are normalizing constants, and λ_0 and λ_1 are parameters specifying the level of noise in the resistivity measurements. Small values of λ_0 and λ_1 reflect noisy resistivity data, while higher of λ_0 and λ_1 reflect less noisy resistivity data. Essentially, the auxiliary variable u_i^j can be interpreted as a noisy realisation of the saturation value m_i^j. The logic behind the choice of distributions in Eqs. (8) and (9) is that it should be more likely to generate u_i^j-values in the correct interval and less likely to generate u_i^j-values in the wrong interval; for example, given that $k_i^j = 0$, it should be more likely to generate u_i^j-values in $[s_{wi}, r]$ than in $(r, 1]$. Moreover, the distributions in Eqs. (8) and (9) ensure that it becomes more and more unlikely to generate u_i^j-values the further you step away from the correct interval. The parameters λ_0 and λ_1 determine how fast this decrease in probability occurs and thereby the spread in the u_i^j-values. The spread in the u_i^j-values, in turn, controls the spread, or the level of noise, in the corresponding d_i^j-values obtained from Eq. (6). If λ_0 and λ_1 are relatively large, most of the generated u_i^j-values will be located within the correct intervals, which in turn results in d_i^j-values with relatively little noise. Likewise, if λ_0 and λ_1 are relatively small, many of the generated u_i^j-values will be located outside the correct intervals, which results in a larger spread in the d_i^j-values and hence more noise. The left plot in Fig. 1 shows $p_0(u_i^j)$ and $p_1(u_i^j)$ when $\lambda_0 = 9.8$, $\lambda_1 = 5$ and $r = 0.3$. Combining that d_i^j is a transformation of u_i^j as given in Eq. (7) and that the distribution for u_i^j is as specified in Eqs. (8) and (9), the likelihood model $p(d_i^j | k_i^j)$ can be derived. When $a = b = 2$ it can be shown that

$$\begin{aligned} & p(d_i^j | k_i^j = 0) \\ & = \begin{cases} \frac{c_0 \sqrt{R_w}}{2\phi^j} \left(d_i^j\right)^{-3/2} \exp\left\{-\frac{\lambda_0}{\phi^j}\sqrt{\frac{R_w}{d_i^j}}\right\} & \text{when } d_i^j \in \left(\frac{R_w}{(\phi^j)^2}, \frac{R_w}{r^2(\phi^j)^2}\right], \\ \frac{c_0 \sqrt{R_w}}{2\phi^j} \left(d_i^j\right)^{-3/2} \exp\left\{-\lambda_0 r\right\} & \text{when } d_i^j \in \left(\frac{R_w}{r^2(\phi^j)^2}, \frac{R_w}{s_w^2(\phi^j)^2}\right] \end{cases} \end{aligned} \tag{10}$$

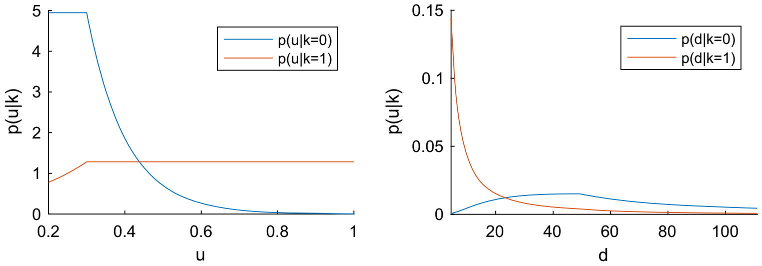


Fig. 1 Likelihood model: Plots of Eqs. (8) and (9) to the left, and Eqs. (10) and (11) to the right, when $\phi = 0.15$, $R_w = 0.1$, $\lambda_0 = 9.8$ and $\lambda_1 = 5$

and

$$p(d_i^j|k_i^j = 1) = \begin{cases} \frac{c_1\sqrt{R_w}}{2\phi^j}\left(d_i^j\right)^{-3/2} \exp\{\lambda_1 r\} & \text{when } d_i^j \in \left(\frac{R_w}{(\phi^j)^2}, \frac{R_w}{r^2(\phi^j)^2}\right], \\ \frac{c_1\sqrt{R_w}}{2\phi^j}\left(d_i^j\right)^{-3/2} \exp\left\{\frac{\lambda_1}{\phi^j}\sqrt{\frac{R_w}{d_i^j}}\right\} & \text{when } d_i^j \in \left(\frac{R_w}{r^2(\phi^j)^2}, \frac{R_w}{s_w^2(\phi^j)^2}\right]. \end{cases} \tag{11}$$

The right plot in Fig. 1 shows $p(d_i^j|k_i^j = 0)$ and $p(d_i^j|k_i^j = 1)$ when $\lambda_0 = 9.8$, $\lambda_1 = 5, a = b = 2, R_w = 0.1, \phi^j = 0.15$ and $r = 0.3$.

Combining the estimated prior Markov chain for column c with the likelihood model specified above, the corresponding posterior distribution also becomes a non-stationary Markov chain. The properties of this posterior Markov chain are computationally easy to compute, and in particular the bivariate distributions for every two neighbor nodes in column c can be found. To update the prior ensemble members of column c, a distribution $q\left(\tilde{k}_{i,c}^{(l)}\middle| k_{i,c}^{(l)}\right)$ which preserves these bivariate distributions is constructed. More specifically, under the assumption that the estimated prior Markov chain for column c is correct, $k_{i,c}^{(l)}$ is updated by simulating from a conditional distribution $q\left(\tilde{k}_{i,c}^{(l)}\middle| k_{i,c}^{(l)}\right)$ such that the bivariate distribution for every pair of neighbor nodes in $\tilde{k}_{i,c}^{(l)}$ is equal to the corresponding bivariate distribution of the posterior Markov chain for column c.

The chosen distribution $q\left(\tilde{k}_{i,c}^{(l)}\middle| k_{i,c}^{(l)}\right)$ for updating the prior ensemble members of column c can be expressed as

$$q\left(\tilde{k}_{i,c}^{(l)}\middle| k_{i,c}^{(l)}\right) = q_1\left(\tilde{k}_{i,(1,c)}^{(i)}\middle| k_{i,(1,c)}^{(l)}\right)\prod_{s=2}^{S} q_s\left(\tilde{k}_{i,(s,c)}^{(l)}\middle| \tilde{k}_{i,(s-1,c)}^{(l)}, k_{i,(s,c)}^{(l)}\right), \tag{12}$$

where S is the number of rows in the lattice used to represent k_i, and $k_{i,(s,c)}^{(l)}$ and $\tilde{k}_{i,(s,c)}^{(l)}$ are the s'th elements in column c of $k_i^{(l)}$ and $\tilde{k}_i^{(l)}$, respectively. Thus, given the prior ensemble member $k_{i,c}^{(l)}$, the distribution of the posterior ensemble member $\tilde{k}_{i,c}^{(l)}$ is a Markov chain with initial distribution specified by $q_1(\cdot|\cdot)$ and transition probabilities specified by $q_s(\cdot|\cdot,\cdot), s = 2,\ldots,S$. To specify the updating procedure completely it now remains to specify $q_1(\cdot|\cdot)$ and $q_s(\cdot|\cdot,\cdot), s = 2,\ldots,S$. These are specified to accomplish two goals. First, considering $k_{i,c}^{(l)}$ as a sample from the estimated prior Markov chain, the marginal bivariate distributions for every pair $\left(\tilde{k}_{i,(s-1,c)}^{(l)}, \tilde{k}_{i,(s,c)}^{(l)}\right), s = 2,\ldots,S$ should be identical to the corresponding bivariate distribution in the posterior Markov chain discussed above. This requirement ensures that the updated fluid-facies values reflect the new resistivity data d_i. However, with only this requirement many possible solutions exist for $q_1(\cdot|\cdot)$ and $q_s(\cdot|\cdot,\cdot), s = 2,\ldots,S$, so there is room for formulating another goal. Still considering $k_{i,c}^{(l)}$ as a sample from the estimated prior Markov chain, the second goal for the updating of $k_{i,c}^{(l)}$ is to maximise the expected number of elements in $k_{i,c}^{(l)}$ that remain unchanged; that is, the goal is to maximize

$$\mathrm{E}\left[\sum_{s=1}^{S} I\left(k_{i,(s,c)}^{(l)} = \tilde{k}_{i,(s,c)}^{(l)}\right)\right], \tag{13}$$

where $I(\mathscr{A})$ equals one if the event $\mathscr{A}$ is true, and zero otherwise, and the expectation is taken with respect to the joint distribution of $k_{i,c}^{(l)}$ and $\tilde{k}_{i,c}^{(l)}$. This requirement makes the updating robust with respect to the a priori Markov chain assumption made for $k_{i,c}^{(l)}, l = 1,\ldots,M$. If the true distribution of $k_{i,c}^{(l)}, l = 1,\ldots,M$ is not a Markov chain, many of its non-Markov properties will prevail into $\tilde{k}_{i,c}^{(l)}, l = 1,\ldots,M$ since it is specified that as few changes as possible should be made to $k_{i,c}^{(i)}$ in the generation of $\tilde{k}_{i,c}^{(l)}$. Numerically, it turns out that in that the maximization of the expression in Eq. (13) under the constraints for the specified bivariate distributions for the pairs $\left(\tilde{k}_{i,(s-1,c)}^{(l)}, \tilde{k}_{i,(s,c)}^{(l)}\right), s = 2,\ldots,S$ can be efficiently computed using a combination of dynamic programming and linear programming. The details of the optimization algorithm are discussed in Loe and Tjelmeland (2020).

2.3.2 The Forecast Step

The forecast step of the proposed approach also involves two parts. First, the ensemble $\left\{\tilde{k}_i^{(1)},\ldots,\tilde{k}_i^{(M)}\right\}$ is mapped to a corresponding ensemble $\left\{\tilde{m}_i^{(1)},\ldots,\tilde{m}_i^{(M)}\right\}$. Second, the forward model in Eq. (2) is used to generate $\left\{m_{i+1}^{(1)},\ldots,m_{i+1}^{(M)}\right\}$ from $\left\{\tilde{m}_i^{(1)},\ldots,\tilde{m}_i^{(M)}\right\}$. In the following, the two parts of the forecast step are discussed in more detail.

To generate the saturation field $\tilde{m}_i^{(l)}$ based on a given fluid-facies field $\tilde{k}_i^{(l)}$, the fluid-facies indicators in $\tilde{k}_i^{(l)}$ are first used to define a lattice of distances, δ, from each node j to a node with the opposite value of node j. More precisely, the values in δ are

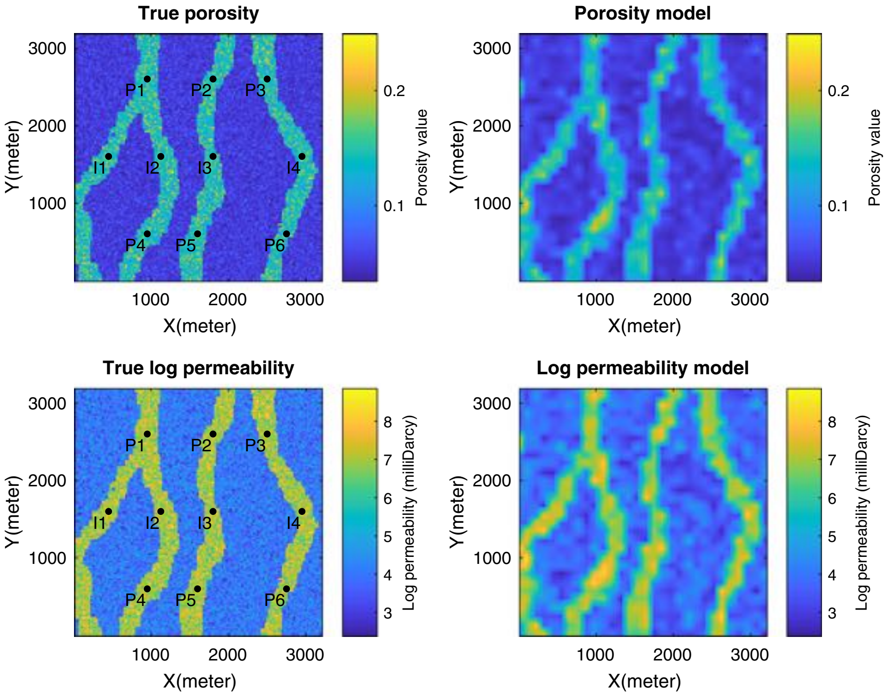


Fig. 2 To the left: True porosity and log permeability models, with the locations of the production and injection wells marked P and I, respectively. To the right: Assumed porosity and log permeability models

defined sequentially as follows. First, $\delta^j = 0$ is set for all nodes j that has one or more neighbor node j' so that $\tilde{k}_i^{(l),j} \neq \tilde{k}_i^{(l),j'}$. Thereafter, $\delta^j = 1$ is set for all nodes j for which δ^j is still undefined and which has a neighbor node j' with $\delta^{j'} = 0$. Thereafter, $\delta^j = 2$ is set for all nodes j for which δ^j is still undefined and which has a neighbor node j' with $\delta^{j'} = 1$. This process is continued until δ^j is defined for all nodes j. The next step is to scale the δ^j values into the [0, 1] interval. Letting Δ denote the scaled field, the value for node j is defined as

$$\Delta^j = \begin{cases} 0 & \text{if } \delta^j > \delta_{\max} \text{ and } \tilde{k}_i^{(l),j} = 0, \\ \frac{1}{2} - \frac{\delta^j + \frac{1}{2}}{2\delta_{\max}+1} & \text{if } \delta^j \leq \delta_{\max} \text{ and } \tilde{k}_i^{(l),j} = 0, \\ \frac{1}{2} + \frac{\delta^j + \frac{1}{2}}{2\delta_{\max}+1} & \text{if } \delta^j \leq \delta_{\max} \text{ and } \tilde{k}_i^{(l),j} = 1, \\ 1 & \text{if } \delta^j > \delta_{\max} \text{ and } \tilde{k}_i^{(l),j} = 1, \end{cases} \tag{14}$$

where $\delta_{\max} > 0$ is a parameter controlling the size of the transition zone from s_{wi} to 1. The larger the value of δ_{max}, the larger the size of the transition zone. One should choose a value for δ_{max} based on what one believes is a realistic transition for the application in consideration. In the numerical examples in Sect. 3, $\delta_{\max} = 8$ is used. The Δ field defines a trend for the $\tilde{m}_i^{(l)}$ values. To add noise to this trend a slightly modified version of the so-called smootherstep function is first used to transform the

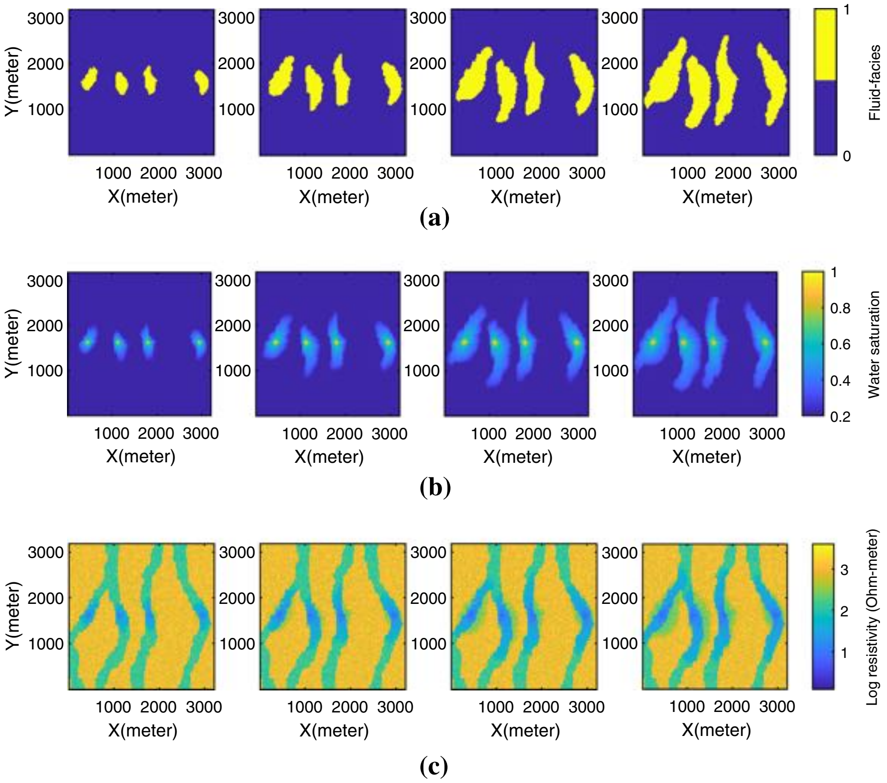


Fig. 3 Reference model: **a** Fluid-facies k_i, **b** water saturation values m_i and **c** log resistivity values d_i at time steps (from left to right) $i = 6, 12, 18$ and 24

Δ^j values over to the real line,

$$\nu^j = 2\Phi^{-1}(0.99999)\left[6\left(\Delta^j\right)^5 - 15\left(\Delta^j\right)^4 + 10\left(\Delta^j\right)^3\right] \\ -\Phi^{-1}(0.99999) + \Phi^{-1}(r), \tag{15}$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard normal distribution and $\Phi^{-1}(\cdot)$ is the inverse of this function. The effect of Eq. (15) is that the value ν^j is in the left tail of a normal distribution with mean $\Phi^{-1}(r)$ and unit variance when $\Delta^j = 0$, or in the right tail of the same distribution when $\Delta^j = 1$, and with a smooth transition between these two extremes. Moreover, the last term in Eq. (15) ensures that $\nu^j = \Phi^{-1}(r)$ when $\Delta^j = 0.5$. A noisy version z of ν is then defined by setting

$$z^j = \sqrt{1-\alpha^2}(\nu^j - \Phi^{-1}(r)) + \alpha\varepsilon^j, \tag{16}$$

Table 1 Experimental settings for the three case studies

	# of measurements	Measurement noise	Likelihood model parameters
Case 1	24 (every 6 months)	Low	$\lambda_0 = 9.8, \quad \lambda_1 = 5.0$
Case 2	4 (every 3 years)	Low	$\lambda_0 = 9.8, \quad \lambda_1 = 5.0$
Case 3	4 (every 3 years)	High	$\lambda_0 = 7.8, \quad \lambda_1 = 2.5$

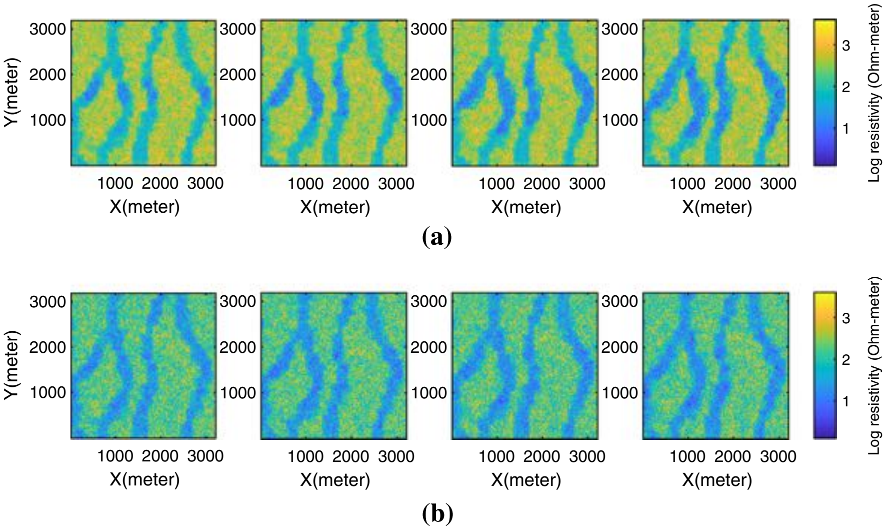


Fig. 4 Log resistivity observations d_i at time steps (from left to right) $i = 6, 12, 18$ and 24, in **a** for cases 1 and 2, and in **b** for case 3

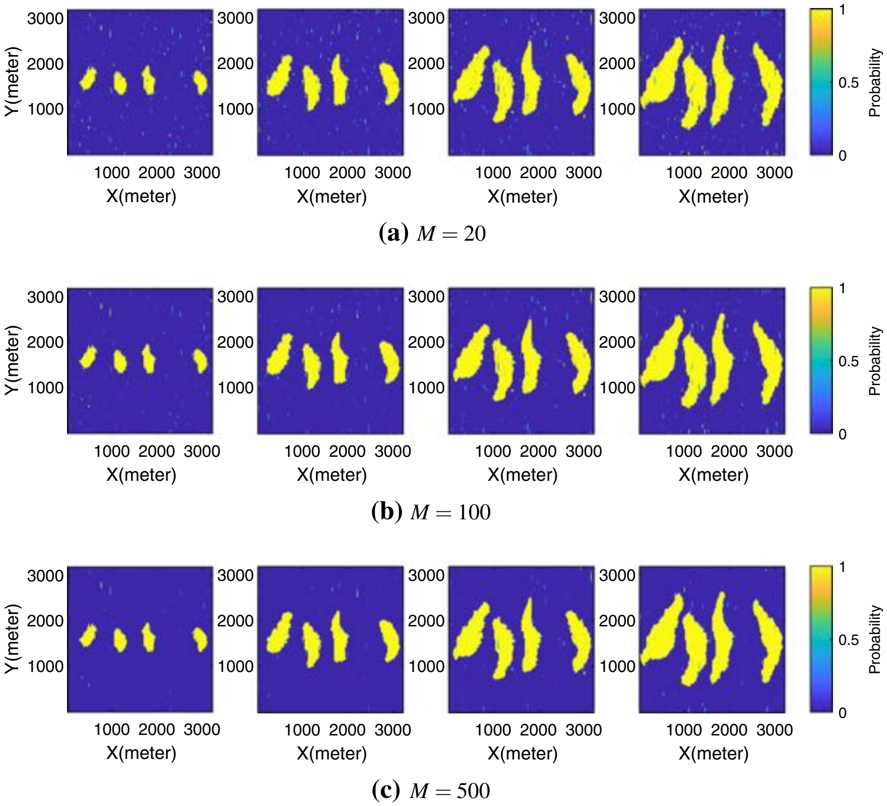
where ε is a Gaussian field with zero mean, unit variance and an exponential correlation function, and $\alpha > 0$ is a parameter controlling the noise level. Finally, the saturation field is defined by transforming the z field back to the $(s_{wi}, 1)$ interval,

$$\tilde{m}_t^{(i),j} = s_{wi} + (1 - s_{wi})\Phi \left(z^j + \Phi^{-1} \left(\frac{r - s_{wi}}{1 - s_{wi}} \right) \right). \tag{17}$$

The second part in the forecast step, to generate the ensemble $\left\{ m_{i+1}^{(1)}, \ldots, m_{i+1}^{(M)} \right\}$ from the ensemble $\left\{ \tilde{m}_i^{(1)}, \ldots, \tilde{m}_i^{(M)} \right\}$, is simply done by using the forward model in Eq. (2) for each ensemble member separately; that is, by setting

$$m_{i+1}^{(l)} = g\left(\tilde{m}_i^{(l)} \right) \tag{18}$$

for $l = 1, \ldots, M$.



(a) $M = 20$

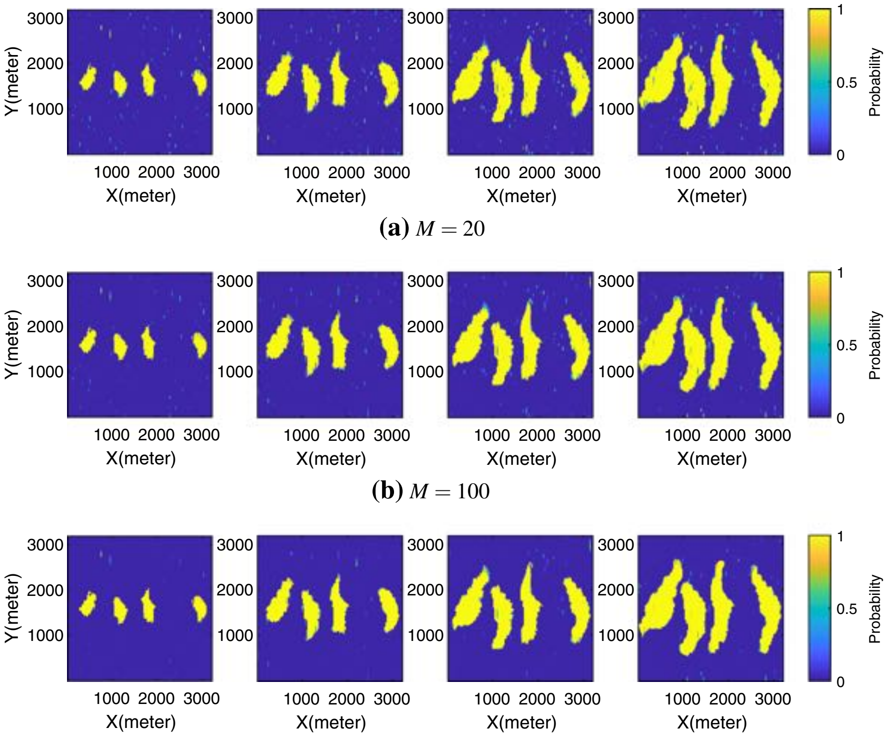
(b) $M = 100$

(c) $M = 500$

Fig. 5 Results from case 1: Images of estimated marginal probabilities $p(k_i^j = 1|d_{1:i})$ at time steps (from left to right) $i = 6$, $i = 12$, $i = 18$, and $i = 24$ using three different ensemble sizes

3 Application

The proposed inversion method is tested using a synthetic reservoir model. The model consists of a two-dimensional reservoir, 25 m × 25 m, with constant thickness and four main channels with high-porosity rocks surrounded by low-porosity rocks; see Fig. 2. The fluid system includes two fluid phases: oil and water. Therefore, two fluid-facies are defined: oil-saturated rocks (corresponding to the value 0) and water-saturated rocks (corresponding to the value 1). The discretized reservoir is defined on a 128×128 grid, and the well configuration includes four injectors and six producers as shown in Fig. 2. The oil production mechanism is based on water injection simulated using the MATLAB Reservoir Simulation Toolbox (Lie 2019) for a time period of 12 years. The 12 year time period is discretized into 24 equidistant time points $t_1, \ldots, t_{24}$ such that each step of the simulation involves propagating the system six months forward in time. During the simulation, injection rates are kept constant at the injector locations, and bottom hole pressure is kept constant at the producer locations. Initially, the entire reservoir is filled with hydrocarbon with the irreducible water saturation value of



(a) $M = 20$

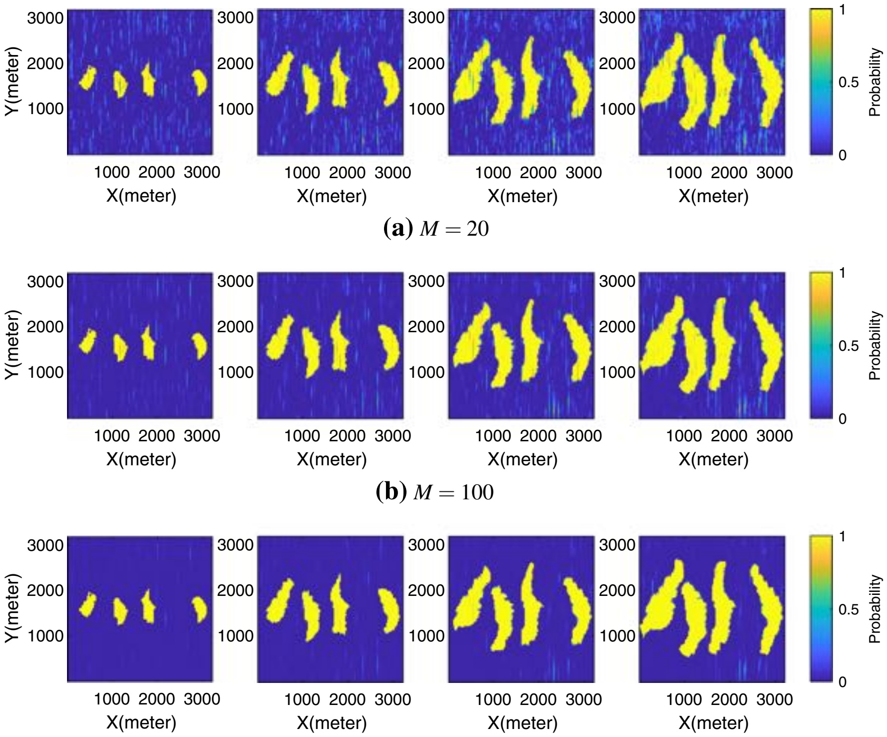
(b) $M = 100$

(c) $M = 500$

Fig. 6 Results from case 2: Images of estimated marginal probabilities $p(k_i^j = 1|d_{1:i})$ at time steps (from left to right) $i = 6$, $i = 12$, $i = 18$, and $i = 24$ using three different ensemble sizes

$s_{wi} = 0.2$. Based on a preliminary analysis, the threshold parameter r in Eq. (3) is set to $r = 0.3$. Figure 3a, b show the fluid-facies k_i and saturation values m_i, respectively, in the reservoir at the time steps $i = 6$, $i = 12$, $i = 18$ and $i = 24$; that is, after $t_6 = 3$, $t_{12} = 6$, $t_{18} = 9$ and $t_{24} = 12$ years of the simulation. Figure 3c shows corresponding reference resistivity values (in log-scale); that is, the resistivity values one obtains by inserting the true water saturation values into Archie's law in Eq. (4). Pretending that the fluid-facies and saturation values used to generate the plots in Fig. 3 are unknown, the goal of the simulation experiment is to estimate the fluid-facies field at each time step based on noisy resistivity data. In this example, the resistivity data d_i at time t_i includes a two-dimensional map of resistivity measurements; specifically, the dimensionality N_d of d_i is equal to the dimensionality N_k of k_i (and m_i) so that an observation d_i^j is available for each variable k_i^j of k_i. Since a 128×128 grid is considered, with a fluid-facies variable k_i^j in every cell j, the dimensions N_k and N_d are $N_k = N_d = 128 \cdot 128 = 16384$.

The porosity and permeability models shown in Fig. 2 (left plots) are the true porosity and permeability models of the reservoir. These were the values used to generate the reference model shown in Fig. 3. Since porosity and permeability are



(a) $M = 20$

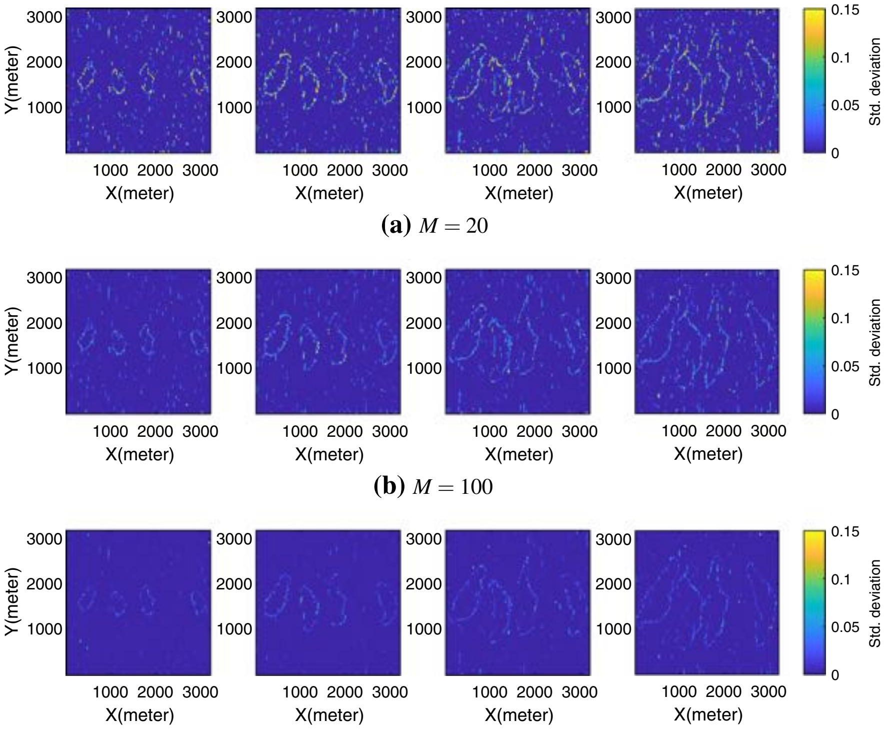
(b) $M = 100$

(c) $M = 500$

Fig. 7 Results from case 3: Images of estimated marginal probabilities $p(k_i^j = 1|d_{1:i})$ at time steps (from left to right) $i = 6$, $i = 12$, $i = 18$, and $i = 24$ using three different ensemble sizes

generally not known, a reservoir model of assumed porosity and permeability models is built to mimic the resolution of a reservoir model estimated from pre-production seismic data. The assumed porosity and permeability models are shown in Fig. 2 (right plots).

Three case studies are presented, differing in the frequency with which the resistivity measurements are collected and the amount of noise in the measurements; see Table 1. The first case, referred to as case 1, represents an idealized situation where resistivity measurements are recorded frequently and the degree of noise in the data is small. Specifically, observations are assumed to be recorded every six months, or at every time step $i = 1, \ldots, N_t$. Hence the set $\mathbb{T}$ introduced in Sect. 2.1 is $\mathbb{T} = \{1, 2, \ldots, 24\}$. Figure 4a shows the simulated resistivity measurements d_i (in log-scale) at the four time steps $i = 6, 12, 18$ and 24 for case 1. The resistivity data were generated with the likelihood model specified in Sect. 2, using the true fluid-facies shown in Fig. 3a and the assumed porosity model shown to the right in Fig. 2, and with the parameters λ_0 and λ_1 set to $\lambda_0 = 9.8$ and $\lambda_1 = 5$. These values for λ_0 and λ_1 represent optimistic noise conditions. In the second case, referred to as case 2, the same data as in case 1 are considered, but observations are assumed to be acquired only every three years of the simulation period; that is, an observation is recorded after 3, 6, 9, and 12 years, or at



(a) $M = 20$

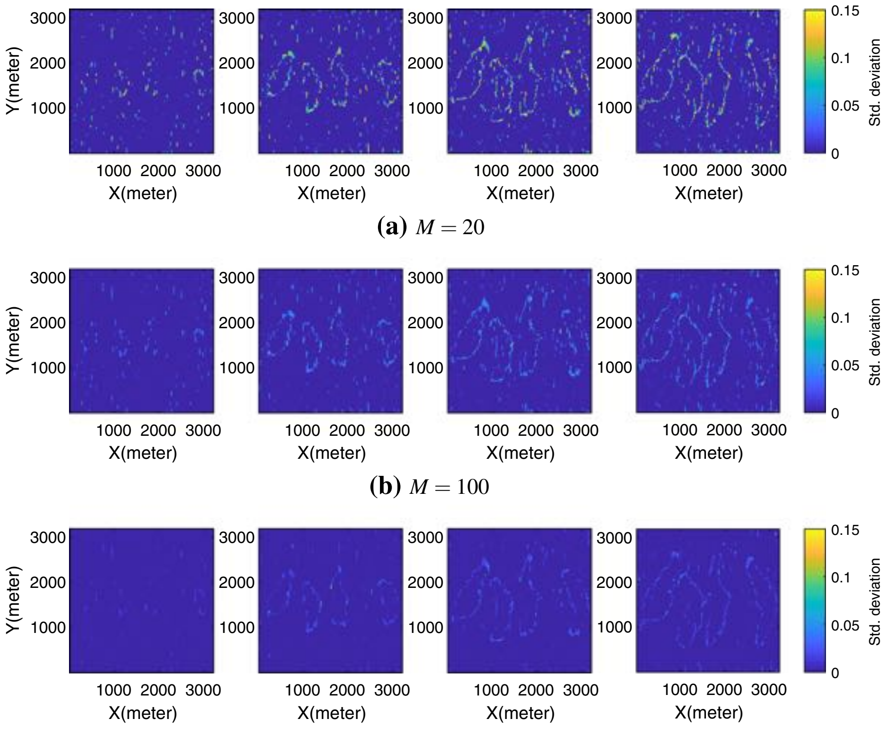
(b) $M = 100$

(c) $M = 500$

Fig. 8 Results from case 1: Empirical standard deviations of $\{\hat{p}^{M}_{A,1}(k^j_i = 1|d_{1:i})\}^{10}_{A=1}$ at time steps (from left to right) $i = 6$, $i = 12$, $i = 18$ and $i = 24$ using three different ensemble sizes

the time steps $i = 6, 12, 18$ and 24. Hence the set $\mathbb{T}$ is in this case $\mathbb{T} = \{6, 12, 18, 24\}$, and the likelihood parameters λ_0 and λ_1 are the same as in case 1. In the third case, referred to as case 3, observations, as in case 2, are acquired only every 3 years, but a different set of data with a much higher level of noise is considered. Hence this case represents the most realistic of the three cases. Figure 4b shows the simulated resistivity measurements for case 3. Similarly to the resistivity data for cases 1 and 2, the resistivity measurements for case 3 were generated using the likelihood model specified in Sect. 2, but with the parameters λ_0 and λ_1 set to $\lambda_0 = 7.8$ and $\lambda_1 = 2.5$. These parameter values represent realistic noise conditions.

For all three case studies the proposed inversion method is tested using three different ensemble sizes: $M = 20$, $M = 100$ and $M = 500$. The parameters $\delta_{\max}$ and α in Eqs. (14) and (16) are set to $\delta_{\max} = 8$ and $\alpha = 0.2$. The ensembles were initialised by first introducing an initial field of fluid facies k_0 for which it is assumed that $k^j_0 = 0$ for every cell j in the reservoir, and thereafter generate each $m^{(l)}_1$ from k_0 as discussed in Sect. 2.3.2. To evaluate the results, an estimate $\hat{p}(k^j_i = 1|d_{1:i})$ for each marginal probability $p(k^j_i = 1|d_{1:i})$, $j = 1, \ldots, N_k$, is computed; specifically, each $p(k^j_i = 1|d_{1:i})$ is estimated as the fraction of updated k^j_i-samples equal to one.



(a) $M = 20$

(b) $M = 100$

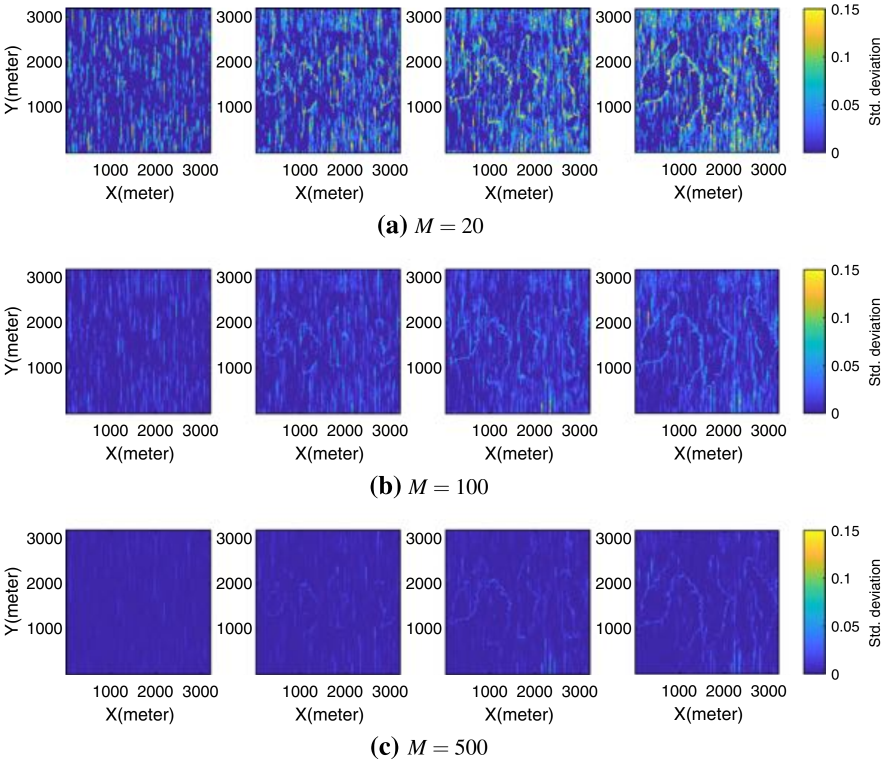
(c) $M = 500$

Fig. 9 Results from case 2: Empirical standard deviations of $\{\hat{p}^M_{A,2}(k_i^j = 1|d_{1:i})\}_{A=1}^{10}$ at time steps (from left to right) $i = 6$, $i = 12$, $i = 18$ and $i = 24$ using three different ensemble sizes

Mathematically, that is

$$\hat{p}(k_i^j = 1|d_{1:i}) = \frac{1}{M}\sum_{l=1}^{M} \tilde{k}_i^{j,(l)}. \tag{19}$$

Figures 5, 6 and 7 present images of these estimated marginal probabilities for cases 1, 2 and 3, respectively. Comparison between the plots in Figs. 5 to 7 and the reference k_i-values in Fig. 3a shows that the proposed inversion method to a large extent has captured the true underlying binary field of fluid-facies in all three cases, even when using the small ensemble size $M = 20$. As expected, larger ensemble sizes provide more accurate results, but the results obtained with $M = 20$ are also satisfactory. However, a few short vertical lines have a tendency to appear in some of the figures, especially in the results from case 3 in Fig. 7. This is an inevitable spatial effect due to the columns of the grid being updated independently of each other in the conditioning step of the inversion method. It is reasonable that the effect is more apparent in the results from case 3 than in the results from cases 1 and 2, since the quality of the geophysical data in case 3 is lower (i.e., noisier).



(a) $M = 20$

(b) $M = 100$

(c) $M = 500$

Fig. 10 Results from case 3: Empirical standard deviations of $\{\hat{p}^M_{A,3}(k^j_i = 1|d_{1:i})\}^{10}_{A=1}$ at time steps (from left to right) $i = 6$, $i = 12$, $i = 18$ and $i = 24$ using three different ensemble sizes

To evaluate how sensitive the proposed inversion method is on the ensemble size M, ten independent runs are performed for each of the ensemble sizes $M = 20$, $M = 100$ and $M = 500$ in each of the three cases. Thereafter the marginal probabilities $p(k^j_i = 1|d_{1:i})$, $j = 1, \ldots, N_k$ are estimated cf. Eq. (19). Thereby, in each of the three cases, ten different estimates $\hat{p}(k^j_i = 1|d_{1:i})$ of $p(k^j_i = 1|d_{1:i})$ are obtained for each ensemble size. In the following, let $\hat{p}^M_{A,B}(k^j_i|d_{1:i})$ denote the estimate of $p(k^j_i|d_{1:i})$ obtained in run number $A = 1, \ldots, 10$ of case number $B = 1, 2, 3$ when using ensemble size $M = 20, 100, 500$. To evaluate the accuracy of the ten estimates $\hat{p}^M_{1,B}(k^j_i|d_{1:i}), \ldots, \hat{p}^M_{10,B}(k^j_i|d_{1:i})$ of $p(k^j_i = 1|d_{1:i})$ obtained in the ten runs of case B when using ensemble size M, the standard deviation of these ten estimates is computed. Results are shown in Figs. 8, 9 and 10 for cases 1, 2 and 3, respectively. Similarly to the other results presented above, the results obtained with the higher ensemble sizes $M = 100$ and $M = 500$ are overall smoother and less noisy than those obtained with the rather small ensemble size $M = 20$. A general trend, however, for all three ensemble sizes and all three cases, is that the standard deviations tend to be higher near the boundary of the fluid front, which is reasonable, since this is the most uncertain area where changes occur. Moreover, the results from case 3 in Fig. 10 are considerably noisier than the results from cases 2 and 3 in Figs. 8 and 9. This means that the case 3 results tend to vary more from one run to another. Again, this is reasonable, since the resistivity measurements in case 3 are more uncertain.

4 Conclusions

A novel method for monitoring and updating the evolution of fluid-facies from time-lapse geophysical properties in a two-phase flow problem has been presented. The inversion method is based on an ensemble filtering method where the updating of the prior ensemble at each time step is performed using a particular updating method for binary vectors. The main novelty of the work is the extension of ensemble-based methods to mixed discrete-continuous problems to update the spatial distribution of fluid-facies. In the proposed application, the geophysical dataset includes time-lapse resistivity values that are assumed to have been estimated from CSEM data through a preliminary inversion process. The proposed method is tested in a synthetic example with a two-dimensional reservoir model. The results from this synthetic example are accurate and support the validation of the proposed methodology. In real data applications, the accuracy of the results depends on the quality of the data in terms of resolution and signal-to-noise ratio, and also on the accuracy of the fluid flow simulator. The main limitation of this work is that uncertainty in the estimation of porosity and permeability are not taken into account. Future research directions aim to extend the proposed method so that porosity and permeability are also treated as random variables and so that the geophysical dataset includes measured data such as electromagnetic amplitude and phase.

Funding Open Access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital).

References

Aki K, Richards PG (1980) Quantitative seismology: theory and methods. W.H. Freeman and Co, New York

Aziz K (1979) Petroleum reservoir simulation, vol 476. Applied Science Publishers, London

Bear J (2013) Dynamics of fluids in porous media. Courier Corporation, Chelmsford

Bergmann P, Schmidt-Hattenberger C, Labitzke T, Wagner FM, Just A, Flechsig C, Rippe D (2017) Fluid injection monitoring using electrical resistivity tomography—five years of CO2 injection at Ketzin, Germany. Geophys Prospect 65(3):859–875

Berre I, Lien M, Mannseth T (2011) Identification of three-dimensional electric conductivity changes from time-lapse electromagnetic observations. J Comput Phys 230(10):3915–3928

Bhuyian AH, Landrø M, Johansen SE (2012) 3D CSEM modeling and time-lapse sensitivity analysis for subsurface CO2 storage. Geophysics 77(5):E343–E355

Buland A, Kolbjørnsen O (2012) Bayesian inversion of CSEM and magnetotelluric data. Geophysics 77(1):E33–E42

Claes N, Paige GB, Grana D, Parsekian AD (2020) Parameterization of a hydrologic model with geophysical data to simulate observed subsurface return flow paths. Vadose Zone J 19(1):e20024

Commer M, Doetsch J, Dafflon B, Wu Y, Daley TM, Hubbard SS (2016) Time-lapse 3-D electrical resistance tomography inversion for crosswell monitoring of dissolved and supercritical CO2 flow at two field sites: Escatawpa and Cranfield, Mississippi, USA. Int J Greenhouse Gas Control 49:297–311

Constable S (2010) Ten years of marine CSEM for hydrocarbon exploration. Geophysics 75(5):75A67–75A81

Doucet A, de Freitas N, Gordon N (2001) Sequential Monte Carlo methods in practice. Springer, New York

Doyen P (2007) Seismic reservoir characterization: an earth modelling perspective, vol 2. EAGE Publications, Houten

Doyen P, Psaila D, Astratti D, Kvamme L, Al Najjar N (2000) Saturation mapping from 4-D seismic data in the Statfjord field. Paper OTC 12100 presented at the Offshore Technology Conference, Houston, Texas, USA, 1-4 May

Evensen G (2009) Data assimilation: the ensemble Kalman filter. Springer, Berlin

Flinchum BA, Holbrook WS, Grana D, Parsekian AD, Carr BJ, Hayes JL, Jiao J (2018) Estimating the water holding capacity of the critical zone using near-surface geophysics. Hydrol Process 32(22):3308–3326

Gasperikova E, Hoversten GM (2006) A feasibility study of nonseismic geophysical methods for monitoring geologic CO2 sequestration. Lead Edge 25(10):1282–1288

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning: data mining, inference, and prediction. Springer, Berlin

Kotikian M, Parsekian AD, Paige G, Carey A (2019) Observing heterogeneous unsaturated flow at the hillslope scale using time-lapse electrical resistivity tomography. Vadose Zone J 18(1):1–16

Leeuwenburgh O, Arts R (2014) Distance parameterization for efficient seismic history matching with the ensemble Kalman filter. Comput Geosci 18(3–4):535–548

Lie KA (2019) An introduction to reservoir simulation using MATLAB/GNU Octave. Cambridge University Press, Cambridge

Lien M, Mannseth T (2008) Sensitivity study of marine CSEM data for reservoir production monitoring. Geophysics 73(4):F151–F163

Lien M, Mannseth T, Agersborg R (2014) Assimilation of time-lapse CSEM data for fluid flow monitoring. In: 76th EAGE conference and exhibition-workshops, 1, European Association of Geoscientists & Engineers, pp 1–3

Loe MK, Tjelmeland H (2020) Ensemble updating of binary state vectors by maximising the expected number of unchanged components. Scand J Stat. https://doi.org/10.1111/sjos.12483

MacGregor L (2012) Integrating seismic, CSEM and well log data for reservoir characterization. Lead Edge 31(3):258–265

Martinez WL, Martinez AR (2015) Computational statistics handbook with MATLAB. Chapman and Hall/CRC, Cambridge

Mavko G, Mukerji T, Dvorkin J (2009) The rock physics handbook, 2nd edn. Cambridge University Press, Cambridge

Oliver DS, Reynolds AC, Liu N (2008) Inverse theory for petroleum reservoir characterization and history matching. Cambridge University Press, Cambridge

Orange A, Key K, Constable S (2009) The feasibility of reservoir monitoring using time-lapse marine CSEM. Geophysics 74(2):F21–F29

Shahin A, Key K, Stoffa P, Tatham R (2012) Petro-electric modeling for CSEM reservoir characterization and monitoring. Geophysics 77(1):E9–E20

Trani M, Arts R, Leeuwenburgh O et al (2012) Seismic history matching of fluid fronts using the ensemble Kalman filter. SPE J 18(01):159–171

Tveit S, Bakr SA, Lien M, Mannseth T (2015) Ensemble-based Bayesian inversion of CSEM data for subsurface structure identification. Geophys J Int 201(3):1849–1867

Tveit S, Mannseth T, Park J, Sauvin G, Agersborg R (2020) Combining CSEM or gravity inversion with seismic AVO inversion, with application to monitoring of large-scale CO2 injection. Comput Geosci 24:1201–1220

Viterbi A (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans Inf Theory 13(2):260–269

Weitemeyer K, Constable S, Key K, Behrens J (2006) First results from a marine controlled-source electromagnetic survey to detect gas hydrates offshore Oregon. Geophys Res Lett 33(3):L03304

Zhang Y, Leeuwenburgh O (2017) Image-oriented distance parameterization for ensemble-based seismic history matching. Comput Geosci 21(4):713–731

Paper III

A generalised and fully Bayesian ensemble updating framework

Margrethe Kvale Loe and Håkon Tjelmeland

Technical report

A generalised and fully Bayesian ensemble updating framework

Margrethe Kvale Loe

Department of Mathematical Sciences, Norwegian University of Science and Technology

Håkon Tjelmeland

Department of Mathematical Sciences, Norwegian University of Science and Technology

Abstract

We propose a generalised framework for the updating of a prior ensemble to a posterior ensemble, an essential yet challenging part of ensemble-based filtering methods. The proposed framework is based on a generalised and fully Bayesian view on the traditional ensemble Kalman filter (EnKF). In the EnKF, the updating of the ensemble is based on Gaussian assumptions, whereas in our setup the updating may be based on another parametric family. In addition, we propose to formulate an optimality criterion and to find the optimal update with respect to this criterion. The framework is fully Bayesian in the sense that the parameters of the assumed forecast model are treated as random variables. As a consequence, a parameter vector is simulated, for each ensemble member, prior to the updating. In contrast to existing fully Bayesian approaches, where the parameters are simulated conditionally on all the forecast samples, the parameters are in our framework simulated conditionally on both the data and all the forecast samples, except the forecast sample which is to be updated. The proposed framework is studied in detail for two parametric families: the linear-Gaussian model and the binary hidden Markov model (HMM). For the linear-Gaussian case, we prove that a particular square root filter is optimal with respect to the criterion of minimising the expected Mahalanobis distance between corresponding prior and posterior ensemble members. Simulation examples for both the linear-Gaussian model and the binary HMM are presented. Here, we observe that the proposed square root filter based on the linear-Gaussian model gives a more realistic representation of the uncertainty than the traditional EnKF and that the effect of not conditioning on the forecast sample which is to be updated can be quite remarkable.

1 Introduction

The ensemble Kalman filter (EnKF) (Burgers et al., 1998; Evensen, 2003) is a recursive Monte Carlo algorithm which provides an approximate solution to the statistical filtering problem. The filter has been successfully applied to problems in several fields of the geosciences, including reservoir evaluation, oceanography, and weather forecasting. Although the EnKF relies on a linear-Gaussian assumption about the underlying state-space model, it has shown to work well even in non-linear, non-Gaussian situations, and it also scales well to problems with very high-dimensional state vectors. The EnKF literature is extensive, and several modifications of the traditional scheme, as presented in Burgers et al. (1998), have been proposed and studied. Much of the literature is quite geophysical-oriented with limited focus on the statistical properties of the filter. In recent years, however, the EnKF has gained a lot of attention also from statisticians (e.g., Katzfuss et al., 2016). In the current report, we take a Bayesian perspective on the EnKF and use it to formulate a new and general class of ensemble filtering methods which also includes filtering of categorical variables.

The EnKF alternates between a forecast step and an update step. The main challenge, and the focus of this report, is the update step. The goal of the update step is to condition an ensemble of (approximate) realisations from a prior, or so-called forecast, distribution on new observations so that a new ensemble of (approximate) realisations from the corresponding posterior, or so-called filtering, distribution is obtained. To cope with this issue, the EnKF introduces Gaussian approximations and updates the forecast samples in the form of a linear shift closely related to the linear update of the mean in the traditional Kalman filter (Kalman, 1960). Since the resulting filtering ensemble is obtained from a linear shift of a possibly non-Gaussian forecast ensemble, non-Gaussian properties may have been captured.

An important property of the EnKF linear update is that it implicitly involves the construction of a Gaussian approximation to the forecast distribution.

In practice, only a covariance matrix is estimated. Combined with the assumption that the likelihood model is linear-Gaussian, the Gaussian approximation to the forecast distribution yields a Gaussian approximation to the filtering distribution according to Bayes' rule. Under the assumption that the forecast ensemble contains independent samples from the Gaussian approximation to the forecast model, the linear shift corresponds to conditional simulation from a Gaussian distribution with mean and covariance so that each updated sample marginally is distributed according to the Gaussian approximation to the filtering distribution. Loe and Tjelmeland (2021) present a generalisation of these underlying features of the EnKF and formulate a general class of ensemble updating procedures. The overall idea behind the framework is that more generally another parametric model than the Gaussian can be pursued when constructing an approximation to the forecast distribution. Likewise, another parametric model than the linear-Gaussian can be pursued for the likelihood model. From Bayes' rule, a corresponding approximation to the filtering distribution follows. To update the prior samples, the authors propose to simulate samples from a distribution conditional on the forecast ensemble such that, given that the forecast samples are distributed according to the constructed approximation to the forecast distribution, the updated samples are distributed according to the corresponding approximation to the filtering distribution, which corresponds to the property of the EnKF linear update.

The traditional EnKF algorithm is known to have a tendency to underestimate the variances in the forecast and filtering distributions, and the filter may in some cases even diverge in the sense that the ensemble mean drifts away from the truth. Various modifications have been proposed to correct for these issues, e.g. localisation (Hamill and Whitaker, 2001; Houtekammer and Mitchell, 2001; Ott et al., 2004) and inflation (Anderson & Anderson, 1999). One contributing reason for the unstable behaviour of the EnKF may be that uncertainty about the estimated covariance matrix is not taken into account. That is, prior to the ensemble update, a covariance matrix is estimated, and thereafter the linear update proceeds as if this estimated covariance matrix were correct, which obviously is not really the case even in a true linear-Gaussian situation. Myrseth and Omre (2010) address this issue and propose a Bayesian hierarchical EnKF (HEnKF) algorithm where the mean and the covariance of the Gaussian forecast approximation are treated as random variables with prior distributions selected from the Gaussian conjugate family. Prior to the linear updating of the ensemble, the covariance matrix is then simulated rather than estimated. Myrseth and Omre (2010) present simulation examples where it is observed that their proposed HEnKF algorithm provides more reliable results than the traditional EnKF and that the variance underestimation problem is reduced. An improved version of the HEnKF algorithm is presented in Tsyrulnikov and Rakitko (2017). Other strategies for incorporating parameter uncertainty in the EnKF are proposed in Stroud et al. (2018) and Katzfuss et al. (2020). All studies suggest that taking parameter uncertainty into account is advantageous.

In the present report, we propose a fully Bayesian version of the framework proposed in Loe and Tjelmeland (2021). The framework is fully Bayesian in the sense that the model parameters of the assumed forecast distribution are treated as random variables. While the framework of Loe and Tjelmeland (2021) can be seen as a generalisation of the traditional EnKF, the framework proposed in the present report can be seen as a generalisation of the HEnKF of Myrseth and Omre (2010), with one important modification. In Myrseth and Omre (2010), a covariance matrix is simulated for each ensemble member by simulating from the distribution of the covariance matrix given all the forecast samples. In a more general context, if we denote the parameters of the forecast model by θ and the forecast samples by $x^{(1)}, \ldots, x^{(M)}$, where M is the ensemble size, this would translate to simulating, for each ensemble member, a parameter vector $\theta^{(i)}$ from the distribution of θ given $x^{(1)}, \ldots, x^{(M)}$. In the present report, however, we propose to adopt a Bayesian model from which it follows that also the incoming observation, say y, must be included in the conditioning, while the forecast sample $x^{(i)}$ to be updated must be excluded. In other words, prior to the updating of $x^{(i)}$, we propose in this report to simulate a parameter $\theta^{(i)}$ conditionally on y and $x^{(1)}, \ldots, x^{(i-1)}, x^{(i+1)}, \ldots, x^{(M)}$. Similarly to Loe and Tjelmeland (2021), we investigate the proposed framework in two situations. First, we consider the situation where the chosen forecast and likelihood approximations constitute a linear-Gaussian model, which corresponds to the model assumptions of the EnKF. Second, we consider the situation where the chosen forecast and likelihood approximations constitute a hidden Markov model (HMM) with categorical states. In contrast to Loe and Tjelmeland (2021), where the core focus is on the situation with the HMM, the current report also gives considerable focus to the linear-Gaussian model and the EnKF. In particular, we formulate a class of (fully Bayesian) EnKF algorithms, of which the traditional EnKF and the square root EnKF (Tippett et al., 2003) are special cases.

The remains of the report take the following outline. First, Section 2 provides some background material on state-space models and the EnKF. Next, our generalised ensemble updating framework is presented in Section 3. In Sections 4 and 5, we consider the proposed framework for the two situations outlined above, i.e. in the case of a linear-Gaussian assumed model and a finite state-space HMM, respectively. In Sections 6 and 7, we present simulation examples for the same two cases. Finally, we finish off with a few closing remarks in Section 8.

2 Preliminaries

In this section, we describe state-space models and the related filtering problem in more detail. We also review the EnKF.

2.1 State-space models

A general state-space model consists of two discrete-time stochastic processes: a latent process $\{x^t\}_{t=1}^T$ where $x^t \in \Omega_x \subseteq \mathbb{R}^n$ is an n-dimensional vector, called the state vector at time step t, and an observed process $\{y^t\}_{t=1}^T$, where $y^t \in \Omega_y \subseteq R^m$ is an m-dimensional vector and a partial observation of x^t. The latent x^t-process, usually called the state process, is assumed to evolve in time according to a first-order Markov chain with initial distribution $p_{x^1}(x^1)$ and transition probabilities $p_{x^t|x^{t-1}}(x^t|x^{t-1})$, $t \geq 2$. The joint distribution of $x^{1:T} = (x^1, \dots, x^T)$ can thus be written as

$$p_{x^{1:T}}(x^{1:T}) = p_{x^1}(x^1) \prod_{t=2}^{T} p_{x^t|x^{t-1}}(x^t|x^{t-1}).$$

The observations $y^1, \dots, y^T$ are assumed to be conditionally independent given the states, with y^t depending on $x^{1:T}$ only through x^t. Hence, the joint likelihood for $y^{1:T} = (y^1, \dots, y^T)$ given $x^{1:T}$ can be written as

$$p_{y^{1:T}|x^{1:T}}(y^{1:T}|x^{1:T}) = \prod_{t=1}^{T} p_{y^t|x^t}(y^t|x^t).$$

A graphical illustration of the general state-space model is shown in Figure 1. When the variables of the state vector are categorical, the model is often called an HMM. Following Künsch (2000), the term HMM is in this report reserved for finite state-space state processes, while the term state-space model may refer to either a categorical or a continuous situation.

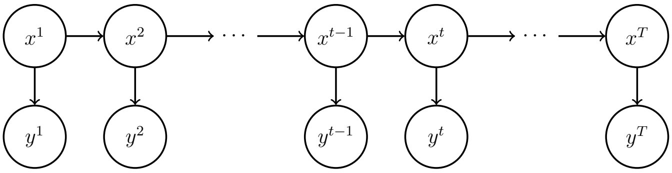


Figure 1: Graphical illustration of a state-space model.

An important inference procedure associated with state-space models, and the main motivation for the work of this report, is filtering. The objective of filtering is, for each time step t, to compute the so-called filtering distribution, $p_{x^t|y^{1:t}}(x^t|y^{1:t})$, that is the distribution of the unobserved state x^t given all the observations available at time t, $y^{1:t} = (y^1, \ldots, y^t)$. Because of the particular dependency structure of the state-space model, the series of filtering distributions can be computed recursively according to a two-step procedure as follows:

$$p_{x^t|y^{1:t-1}}(x^t|y^{1:t-1}) = \int_{\Omega_x} p_{x^t|x^{t-1}}(x^t|x^{t-1}) p_{x^{t-1}|y^{1:t-1}}(x^{t-1}|y^{1:t-1}) \mathrm{d}x^{t-1}, \qquad (1)$$

$$p_{x^t|y^{1:t}}(x^t|y^{1:t}) = \frac{p_{x^t|y^{1:t-1}}(x^t|y^{1:t-1}) p_{y^t|x^t}(y^t|x^t)}{\displaystyle\int_{\Omega_x} p_{x^t|y^{1:t-1}}(x^t|y^{1:t-1}) p_{y^t|x^t}(y^t|x^t) \mathrm{d}x^t}. \qquad (2)$$

The first step is called the prediction step and computes the forecast distribution $p_{x^t|y^{1:t-1}}(x^t|y^{1:t-1})$. The second step is called the update step and uses Bayes' rule to condition the forecast (prior) distribution on the incoming observation y^t to compute the filtering (posterior) distribution $p_{xt|y^{1:t}}(x^t|y^{1:t})$.

Generally, we are unable to evaluate the integrals in Eqs. (1) and (2), and the forecast and filtering distributions are left intractable. Approximate solutions are therefore necessary. The most common approach is to use a simulation-based method where a set of samples is used to empirically represent the series of prediction and filtering distributions. These methods are in the literature often referred to as ensemble methods, and the set of samples used to approximate the distributions is called an ensemble. Starting from an ensemble of independent realisations from the initial model $p_{x^1}(x^1)$, the idea is to advance the ensemble forward in time according to the state-space model dynamics. Similarly to the recursion in Eqs. (1) and (2), an ensemble method may alternate between a forecast step and an update step. Assuming at time t that an ensemble $\{\tilde{x}^{t-1,(1)}, \ldots, \tilde{x}^{t-1,(M)}\}$ of M independent realisations from the previous filtering distribution $p_{x^{t-1}|y^{1:t-1}}(x^{t-1}|y^{1:t-1})$ is available, the forecast step is then carried out by simulating $x^{t,(i)}|\tilde{x}^{t-1,(i)} \sim p_{x^t|x^{t-1}}(\cdot|\tilde{x}^{t-1,(i)})$ independently for each i. This yields a forecast ensemble, $\{x^{t,(1)}, \ldots, x^{t,(M)}\}$, with independent realisations from the forecast distribution $p_{x^t|y^{1:t-1}}(x^t|y^{1:t-1})$. Typically in practical applications, we are able to deal with this forecasting, but to a high computational cost. After the forecast step, the forecast ensemble needs to be updated taking the new observation y^t into account, so that a new filtering ensemble, $\{\tilde{x}^{t,(1)}, \ldots, \tilde{x}^{t,(M)}\}$, with independent realisations from the filtering distribution $p_{x^t|y^{1:t}}(x^t|y^{1:t})$ is obtained. However, in contrast to the prediction step, there is no straightforward way to proceed with this updating. Therefore, ensemble filtering methods require approximations in the update step. In the present report, we propose one such approximate updating method.

There exist two main classes of ensemble filtering methods: particle filters (Gordon et al., 1993; Doucet et al., 2001) and variations of the EnKF. Hybrid versions of these filters have also been proposed (e.g., Frei and Künsch, 2012, 2013). In this report, the focus is on the EnKF, and a brief review of the EnKF follows in the next section.

2.2 The ensemble Kalman filter

The EnKF is an ensemble filtering method which relies on Gaussian approximations. The filter was first introduced in Evensen (1994) and several modifications of the algorithm have been proposed in the literature since then. The variety of EnKF methods can be classified into two main categories, stochastic filters and deterministic filters, differing in whether the updating of the ensemble is carried out in a stochastic or deterministic manner. Deterministic filters are also known as square root filters, and this is the term we use in this report.

To understand the EnKF, consider first a linear-Gaussian model where $x \sim \mathcal{N}(x; \mu, Q)$ and $y|x \sim \mathcal{N}(y; Hx, R)$, $\mu \in \mathbb{R}^n$, $Q \in \mathbb{R}^{n\times n}$, $H \in \mathbb{R}^{m\times n}$, and $R \in \mathbb{R}^{m\times m}$. The posterior model corresponding to this linear-Gaussian model is a Gaussian, $\mathcal{N}(x; \mu^*, Q^*)$, with mean vector $\mu^* \in \mathbb{R}^n$ and covariance matrix $Q^* \in \mathbb{R}^{n\times n}$ analytically available from the Kalman filter equations as

$$\mu^* = \mu + K(y - H\mu) \tag{3}$$

and

$$Q^* = (I_n - KH)Q, \tag{4}$$

respectively, where $I_n \in \mathbb{R}^{n\times n}$ is the $n \times n$ identity matrix and

$$K = QH^\top \left(HQH^\top + R\right)^{-1} \tag{5}$$

is the so-called Kalman gain matrix, where we have introduced the notation $A^\top$ to denote the transpose of a matrix A. Now, suppose $x \sim \mathcal{N}(x; \mu, Q)$ and $\epsilon \sim \mathcal{N}(\epsilon; 0, R)$ are independent random samples, and consider the linear transformation

$$\tilde{x} = x + K(y - Hx + \epsilon). \tag{6}$$

It is then a straightforward matter to show that $\tilde{x}|y$ is distributed according to the Gaussian distribution $\mathcal{N}(x; \mu^*, Q^*)$ with mean μ^* and covariance Q^* given by Eqs. (3) and (4), respectively (e.g., Burgers et al., 1998).

At a given time step t, the EnKF starts by making a linear-Gaussian assumption about the true (unknown) underlying model. Specifically, the forecast samples $x^{t,(1)}, \ldots, x^{t,(M)}$ are assumed to be distributed according to a Gaussian distribution $\mathcal{N}(x^t; \mu^t, Q^t)$ where the parameters μ^t and Q^t are set equal to the sample mean and the sample covariance of the forecast ensemble, and the likelihood model is assumed to be a Gaussian distribution with mean $H^t x^t$ and covariance R^t, $H^t \in \mathbb{R}^{m\times n}$, $R^t \in \mathbb{R}^{m\times m}$. Under the assumption that the assumed linear-Gaussian model is correct we have $x^{t,(i)} \sim N(x^t; \mu^t, Q^t)$ for each i, and the goal is to update $x^{t,(i)}$ so that $\tilde{x}^{t,(i)} \sim N(x^t; \mu^{*t}, Q^{*t})$, where μ^{*t} and Q^{*t} are given by Eqs. (3) and (4), respectively, with a superscript t included in the notations, i.e.

$$\mu^{*t} = \mu^t + K^t(y^t - H^t \mu^t) \tag{7}$$

and

$$Q^{*t} = (I_n - K^t H^t)Q^t, \tag{8}$$

where, similarly, K^t is given by Eq. (5), with a superscript t included, $K^t = Q^t(H^t)^\top \left(H^t Q^t (H^t)^\top + R^t\right)^{-1}$. Stochastic and square root EnKFs obtain this result in different ways. The stochastic EnKF proceeds by simulating $\epsilon^{t,(i)} \sim \mathcal{N}(\epsilon^t; 0, R^t)$ independently for $i = 1, \ldots, M$, and then exploits Eq. (6), which now takes the form

$$\tilde{x}^{t,(i)} = x^{t,(i)} + K^t(y^t - H^t x^{t,(i)} + \epsilon^{t,(i)}). \tag{9}$$

The square root EnKF instead performs a non-random linear transformation of $x^{t,(i)}$,

$$\tilde{x}^{t,(i)} = B^t(x^{t,(i)} - \mu^t) + \mu^t + K^t(y^t - H^t\mu^t), \tag{10}$$

where $B^t \in \mathbb{R}^{n\times n}$ is a solution to the quadratic matrix equation

$$B^tQ^t(B^t)^\top = (I_n - K^tH^t)Q^t. \tag{11}$$

If the underlying state-space model really is linear-Gaussian, the distribution of each updated sample converges to the true (Gaussian) filtering distribution as $M \to \infty$. In all other cases, the update is biased. However, since the posterior ensemble is obtained from a linear shift of a possibly non-Gaussian prior ensemble, non-Gaussian properties of the true prior and posterior models may, to some extent, be captured.

3 Generalised, fully Bayesian updating framework

In this section, we formulate a general class of ensemble updating procedures. Recall from previous sections that the goal is to update an ensemble of prior realisations, $\{x^{t,(1)},\dots,x^{t,(M)}\}$, to a corresponding ensemble of posterior realisations, $\{\tilde{x}^{t,(1)},\dots,\tilde{x}^{t,(M)}\}$, taking the new observation y^t into account. To cope with this task, we propose to separately update each of the $x^{t,(i)}$ samples in the prior ensemble to a corresponding $\tilde{x}^{t,(i)}$ sample in the posterior ensemble, and to base the updating of $x^{t,(i)}$ on an assumed Bayesian model. As mentioned previously in the report, the proposed framework can be viewed as a generalisation of the hierarchical EnKF algorithm of Myrseth and Omre (2010) with the modification that the parameters are simulated in a different way. The key steps of the proposed updating framework are summarised in Algorithm 1.

Algorithm 1: General ensemble updating procedure

```
1. Select the assumed distributions $f_{\theta^t}(\theta^t)$, $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{y^t|x^t}(y^t|x^t)$
   introduced in Section 3.1
2. for $i = 1, \dots, M$ do
       a) Simulate
              $\theta^{t,(i)}|x^{t,-(i)}, y^t \sim f_{\theta^t|x^{t,-(i)},y^t}(\theta^t|x^{t,-(i)}, y^t)$
          as described in Section 3.4
       b) Construct the model $q^*(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^{t,(i)}, y^t)$ specified in Sections 3.2 and 3.3
       c) Simulate
              $\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^{t,(i)}, y^t \sim q^*(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^{t,(i)}, y^t)$
   end
```

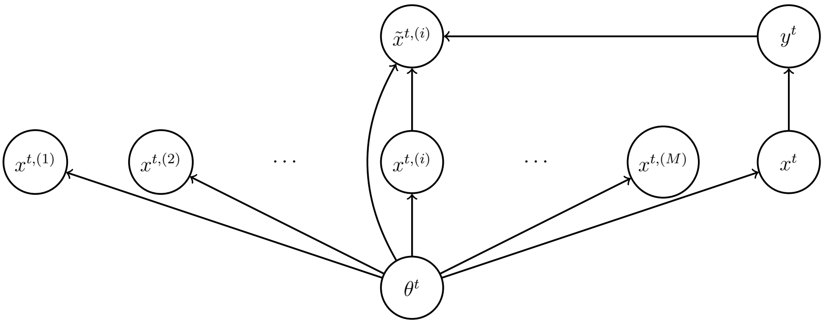


Figure 2: Graphical representation of the assumed Bayesian model for the updating of $x^{t,(i)}$ to $\tilde{x}^{t,(i)}$

3.1 Assumed Bayesian model

For the updating of the forecast sample $x^{t,(i)}$ we adopt an assumed Bayesian model. A graphical illustration of this assumed Bayesian model is shown in Figure 2. The model includes an unknown parameter vector $\theta^t \in \Omega_\theta$, and the forecast samples $x^{t,(1)}, \dots, x^{t,(M)}$ and the latent state vector x^t are assumed to be conditionally independent and identically distributed given θ^t. Moreover, the observation y^t is assumed to be conditionally independent of $x^{t,(1)}, \dots, x^{t,(M)}$ and θ^t given x^t, and the updated sample $\tilde{x}^{t,(i)}$ is restricted to be conditionally independent of x^t and

$$x^{t,-(i)} = \{x^{t,(1)}, \dots, x^{t,(i-1)}, x^{t,(i+1)}, \dots, x^{t,(M)}\}$$

given $x^{t,(i)}$, θ^t and y^t.

To distinguish the assumed Bayesian model from the true and unknown model, we use in the following the notation $f(\cdot)$ to denote distributions associated with the assumed Bayesian model, while, as in previous sections, $p(\cdot)$ is reserved for the truth. Under the assumed Bayesian model, the joint distribution of θ^t, x^t, $x^{t,(1)}, \ldots, x^{t,(M)}$ and y^t reads

$$f_{\theta^t,x^t,x^{t,(1)},\ldots,x^{t,(M)},y^t}(\theta^t, x^t, x^{t,(1)}, \ldots, x^{t,(M)}, y^t) = \\ f_{\theta^t}(\theta^t) f_{x^t|\theta^t}(x^t|\theta^t) f_{y^t|x^t}(y^t|x^t) \prod_{i=1}^{M} f_{x^t|\theta^t}(x^{t,(i)}|\theta^t),$$

where $f_{\theta^t}(\theta^t)$ is an assumed prior model for θ^t, $f_{x^t|\theta^t}(x^t|\theta^t)$ is an assumed prior model for $x^t|\theta^t$ and $f_{y^t|x^t}(y^t|x^t)$ is an assumed likelihood model. The model $f_{x^t|\theta^t}(x^t|\theta^t)$ can be interpreted as an approximation to the intractable forecast model $p_{x^t|y^{1:t-1}}(x^t|y^{1:t-1})$. The model $f_{\theta^t}(\theta^t)$ for θ^t should be chosen as a conjugate prior for $f_{x^t|\theta^t}(x^t|\theta^t)$, while the models $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{y^t|x^t}(y^t|x^t)$ must be chosen so that the corresponding posterior model

$$f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t) \propto f_{x^t|\theta^t}(x^t|\theta^t) f_{y^t|x^t}(y^t|x^t)$$

is tractable.

3.2 Class of updating distributions

Under the assumption that the assumed Bayesian model introduced above is correct, a naïve updating procedure is to sample $\tilde{x}^{t,(i)}$ from $f_{x^t|x^{t,(1)},\ldots,x^{t,(M)},y^t}(x^t|x^{t,(1)}, \ldots, x^{t,(M)}, y^t)$. However, this procedure may be very sensitive to the assumptions of the assumed Bayesian model. To get an updating procedure which is more robust against the assumptions of the assumed model, a better approach is to generate $\tilde{x}^{t,(i)}$ as a modified version of $x^{t,(i)}$ and require

$$f_{\tilde{x}^{t,(i)}|x^{t,-(i)},y^t}(x^t|x^{t,-(i)}, y^t) = f_{x^t|x^{t,-(i)},y^t}(x^t|x^{t,-(i)}, y^t). \tag{12}$$

This way, we use the randomness in $x^{t,(i)}$ to generate randomness in $\tilde{x}^{t,(i)}$, and the forecast sample $x^{t,(i)}$ is therefore not included in the conditioning in Eq. (12). To generate $\tilde{x}^{t,(i)}$ as a modified version of $x^{t,(i)}$ under this restriction, we propose to introduce a distribution $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$ which fulfils Eq. (12), and then simulate $\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t \sim q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$. To construct such a $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$, we first note that the constraint in Eq. (12) can be rewritten as

$$\int_{\Omega_\theta} f_{\theta^t,\tilde{x}^{t,(i)}|x^{t,-(i)},y^t}(\theta^t, x^t|x^{t,-(i)}, y^t)\mathrm{d}\theta^t = \int_{\Omega_\theta} f_{\theta^t,x^t|x^{t,-(i)},y^t}(\theta^t, x^t|x^{t,-(i)}, y^t)\mathrm{d}\theta^t.$$

Using that both x^t and $\tilde{x}^{t,(i)}$ are conditionally independent of $x^{t,-(i)}$ given θ^t and y^t, this can be rewritten as

$$\int_{\Omega_\theta} f_{\theta^t|x^{t,-(i)},y^t}(\theta^t|x^{t,-(i)}, y^t) f_{\tilde{x}^{t,(i)}|y^t,\theta^t}(x^t|y^t, \theta^t)\mathrm{d}\theta^t = \\ \int_{\Omega_\theta} f_{\theta^t|x^{t,-(i)},y^t}(\theta^t|x^{t,-(i)}, y) f_{x^t|y^t,\theta^t}(x^t|y^t, \theta^t)\mathrm{d}\theta^t. \tag{13}$$

A sufficient condition for Eq. (13) to hold is

$$f_{\tilde{x}^{t,(i)}|\theta^t,y^t}(x^t|\theta^t, y^t) = f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t) \tag{14}$$

for all x^t, θ^t, and y^t. Thereby, if for a given θ^t we can manage to construct a $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$ consistent with Eq. (14), we can update $x^{t,(i)}$ by first simulating $\theta^{t,(i)}|x^{t,-(i)}, y^t \sim f_{\theta^t|x^{t,-(i)},y^t}(\theta^t|x^{t,-(i)}, y^t)$ and thereafter simulate $\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^{t,(i)}, y^t \sim q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^{t,(i)}, y^t)$. How to simulate $\theta^{t,(i)}|x^{t,-(i)}, y^t$ is discussed in Section 3.4. To construct a $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^{t,(i)}, y^t)$ consistent with Eq. (14) we note that for $f_{\tilde{x}^{t,(i)}|\theta^t,y^t}(x^t|\theta^t, y^t)$ on the left-hand-side of Eq. (14), we have

$$f_{\tilde{x}^{t,(i)}|\theta^t,y^t}(\tilde{x}^{t,(i)}|\theta^t, y^t) = \int_{\Omega_x} f_{x^t|\theta^t}(x^t|\theta^t) q(\tilde{x}^{t,(i)}|x^t, \theta^t, y^t)\mathrm{d}x^t.$$

Thereby, from Eq. (14), it follows that $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$ must fulfil

$$f_{x^t|\theta^t,y^t}(\tilde{x}^{t,(i)}|\theta^t, y^t) = \int_{\Omega_x} f_{x^t|\theta^t}(x^t|\theta^t) q(\tilde{x}^{t,(i)}|x^t, \theta^t, y^t)\mathrm{d}x^t \tag{15}$$

for all $\tilde{x}^{t,(i)}, \theta^t$ and y^t.

The criterion in Eq. (15) defines a class of updating distributions in the sense that there may be infinitely many distributions $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$ which fulfil Eq. (15). If the assumed model is correct, it does not matter which $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$ within this class we choose; the distribution of $\tilde{x}^{t,(i)}|x^{t,-(i)}, y^t$ equals $f_{x^t|x^{t,-(i)},y^t}(x^t|x^{t,-(i)}, y^t)$ regardless. Generally, however, the assumed model is wrong, and the choice of $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$ can have a substantial effect on the actual distribution of $\tilde{x}^{t,(i)}|x^{t,-(i)}, y^t$. The simplest solution is to set $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$ equal to $f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t)$ which entails that we simulate $\tilde{x}^{t,(i)}$ independently of $x^{t,(i)}$. However, this naïve approach is very sensitive to the assumptions of the assumed model and is not a good way to proceed as we loose a lot of valuable information from $x^{t,(i)}$ about the true (unknown) model that we may not have been able to capture with the assumed model. As discussed above, we want to generate $\tilde{x}^{t,(i)}$ as a modified version of $x^{t,(i)}$. That way, we retain more information from $x^{t,(i)}$ about the true model. An optimal solution $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$ within the class of distributions can be found if an optimality criterion is specified, which we discuss in the next section.

3.3 Optimality criterion

Generally, an optimal solution, denoted $q^*(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$, within the class of distributions defined in the previous section can for example be defined as the solution which minimises the expected value of some function $g(x^{t,(i)}, \tilde{x}^{t,(i)})$,

$$q^*(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t) = \underset{q(\cdot)}{\operatorname{argmin}} \operatorname{E}\left[g(x^{t,(i)}, \tilde{x}^{t,(i)})\right],$$

where the expectation is taken over the distribution $f_{x^t|\theta^t}(x^{t,(i)}|\theta^t)q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$, i.e. the joint distribution of $x^{t,(i)}$ and $\tilde{x}^{t,(i)}$ given (θ^t, y^t) under the assumption that the assumed Bayesian model is correct. In the present report, we propose to choose the function $g(x^{t,(i)}, \tilde{x}^{t,(i)})$ as the Mahalanobis distance between $x^{t,(i)}$ and $\tilde{x}^{t,(i)}$,

$$g(x^{t,(i)}, \tilde{x}^{t,(i)}) = \left(x^{t,(i)} - \tilde{x}^{t,(i)}\right)^\top \Sigma^{-1} \left(x^{t,(i)} - \tilde{x}^{t,(i)}\right), \tag{16}$$

where $\Sigma \in \mathbb{R}^{n \times n}$ is some positive definite matrix. If Σ equals the identity matrix, $g(x^{t,(i)}, \tilde{x}^{t,(i)})$ reduces to the squared Euclidean distance between $x^{t,(i)}$ and $\tilde{x}^{t,(i)}$,

$$g(x^{t,(i)}, \tilde{x}^{t,(i)}) = \sum_{j=1}^{n} \left(x_j^{t,(i)} - \tilde{x}_j^{t,(i)}\right)^2. \tag{17}$$

Basically, the optimality criterion then states that we want to make minimal changes to each prior sample $x^{t,(i)}$. To us, this seems like a reasonable criterion since we want to capture as much information from $x^{t,(i)}$ as possible. Of course, one must value the information that comes with the observation y^t, but there is no reason to make more changes to $x^{t,(i)}$ than necessary.

If x^t is a vector of categorical variables, $x^t_j \in \{0, 1, \ldots, K-1\}$, an alternative is to select $g(x^{t,(i)}, \tilde{x}^{t,(i)})$ as the number of corresponding elements of $x^{t,(i)}$ and $\tilde{x}^{t,(i)}$ that are different,

$$g(x^{t,(i)}, \tilde{x}^{t,(i)}) = \sum_{j=1}^{n} 1\left(x_j^{t,(i)} \neq \tilde{x}_j^{t,(i)}\right), \tag{18}$$

where $1(\cdot)$ denotes the usual indicator function. If each component x^t_j is binary, Eqs. (17) and (18) are equal.

3.4 Parameter simulation

In this section, we describe how to simulate from $f_{\theta^t|x^{t,-(i)},y}(\theta^t|x^{t,-(i)}, y^t)$ when $f_{\theta^t}(\theta^t)$ is chosen as a conjugate prior for $f_{x^t|\theta^t}(x^t|\theta^t)$. Specifically, we can then introduce x^t as an auxiliary variable and simulate (x^t, θ^t) from the joint distribution

$$f_{x^t,\theta^t|x^{t,-(i)},y^t}(x^t, \theta^t|x^{t,-(i)}, y^t) \propto f_{\theta^t}(\theta^t) f_{x^t|\theta^t}(x^t|\theta^t) f_{y^t|x^t}(y^t|x^t) \prod_{j\neq i} f_{x^t|\theta^t}(x^{t,(j)}|\theta^t)$$

by constructing a Gibbs sampler which alternates between drawing x^t from the full conditional distribution $f_{x^t|\theta^t,x^{t,-(i)},y^t}(x^t|\theta^t, x^{t,-(i)}, y^t)$ and θ^t from the full conditional distribution $f_{\theta^t|x^t,x^{t,-(i)},y^t}(\theta^t|x^t, x^{t,-(i)}, y^t)$. Using that x^t and $x^{t,-(i)}$ are conditionally independent given θ^t (see Figure 2), it follows that the full conditional distribution $f_{x^t|\theta^t,x^{t,-(i)},y^t}(x^t|\theta^t, x^{t,-(i)}, y^t)$ is given as

$$f_{x^t|\theta^t,x^{t,-(i)},y^t}(x^t|\theta^t, x^{t,-(i)}, y^t) = f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t).$$

Simulating from $f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t)$ should be achievable, since $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{y^t|x^t}(y^t|x^t)$ are chosen so that $f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t)$ is tractable. Using that θ^t and y^t are conditionally independent given x^t (again, see Figure 2), the other full conditional distribution, $f_{\theta^t|x^t,x^{t,-(i)},y^t}(\theta^t|x^t, x^{t,-(i)}, y^t)$, is given as

$$f_{\theta^t|x^t,x^{t,-(i)},y^t}(\theta^t|x^t, x^{t,-(i)}, y^t) = f_{\theta^t|x^t,x^{t,-(i)}}(\theta^t|x^t, x^{t,-(i)}).$$

Since $f_{\theta^t}(\theta^t)$ is chosen as a conjugate prior for $f_{x^t|\theta^t}(x^t|\theta^t)$, and since x^t, $x^{t,(1)}, \ldots, x^{t,(M)}$ are independent and identically distributed given θ^t, it follows that $f_{\theta^t|x^t,x^{t,-(i)}}(\theta^t|x^t, x^{t,-(i)})$ is tractable and belongs to the same family of distributions as $f_{\theta^t}(\theta^t)$. Simulating from $f_{\theta^t|x^t,x^{t,-(i)}}(\theta^t|x^t, x^{t,-(i)})$ should therefore also be possible.

4 Linear-Gaussian assumed model

In this section, we describe how the general updating procedure described in Section 3 can be applied when the elements of the state vector are continuous variables. As in the EnKF, one may then choose $f_{x^t|\theta^t}(x^t|\theta^t)$ as Gaussian and $f_{y^t|x^t}(y^t|x^t)$ as linear-Gaussian.

4.1 Specification of the assumed model

Suppose $x^t = (x_1^t, \ldots, x_n^t) \in \mathbb{R}^n$ and $y^t = (y_1^t, \ldots, y_m^t) \in \mathbb{R}^m$ are continuous vectors. Let $\theta^t = (\mu^t, Q^t)$ where $\mu^t \in \mathbb{R}^n$, $Q^t \in \mathbb{R}^{n\times n}$, and Q^t is positive definite. Select $f_{x^t|\theta^t}(x^t|\theta^t)$ as a Gaussian distribution with mean vector μ^t and covariance matrix Q^t,

$$f_{x^t|\theta^t}(x^t|\theta^t) = \mathcal{N}(x^t; \mu^t, Q^t),$$

and choose $f_{y^t|x^t}(y^t|x^t)$ as a Gaussian distribution with mean $H^t x^t$, $H^t \in \mathbb{R}^{m\times n}$ and covariance matrix $R^t \in \mathbb{R}^{m\times m}$,

$$f_{y^t|x^t}(y^t|x^t) = \mathcal{N}(y^t; H^t x^t, R^t).$$

For a given θ^t, this model corresponds to the linear-Gaussian model introduced in Section 2.2. The corresponding posterior model $f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t)$ is then a Gaussian distribution with mean vector μ^{*t} and covariance matrix Q^{*t} given by Eqs. (7) and (8), respectively. Following Section 3, we adopt a conjugate prior for θ^t, which in this case entails an inverse Wishart distribution for Q^t,

$$f_{Q^t}(Q^t) = \mathcal{W}^{-1}(Q^t; V, \nu), \tag{19}$$

and a Gaussian distribution for $\mu^t|Q^t$,

$$f_{\mu^t|Q^t}(\mu^t|Q^t) = \mathcal{N}(\mu^t; \mu_0, \kappa^{-1}Q^t), \tag{20}$$

where $\nu, \kappa \in \mathbb{R}$, $\mu_0 \in \mathbb{R}^n$ and $V \in \mathbb{R}^{n\times n}$ are known hyper-parameters.

4.2 Derivation of the class of updating distributions

The restriction in Eq. (14) now entails that the updating distribution $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$ must be chosen so that the integral on the right hand side of Eq. (15) returns a Gaussian distribution with mean vector equal to μ^{*t} in Eq. (7) and covariance matrix equal to Q^{*t} in Eq. (8). To obtain this, we start by selecting $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$ as a Gaussian distribution with mean vector $B^t x^{t,(i)} + C^t y^t + d^t$ and covariance matrix S^t,

$$q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t) = \mathcal{N}\left(\tilde{x}^{t,(i)}; B^t x^{t,(i)} + C^t y^t + d^t, S^t\right), \tag{21}$$

where $B^t \in \mathbb{R}^{n\times n}$, $C^t \in \mathbb{R}^{n\times m}$, $d^t \in \mathbb{R}^n$ and $S^t \in \mathbb{R}^{n\times n}$ are quantities that we need to decide so that Eq. (15) is fulfilled. The B^t, C^t, d^t and S^t can all be functions of θ^t and y^t. From Eq. (21), it follows that the posterior sample $\tilde{x}^{t,(i)}$ can be obtained as a linear shift of $x^{t,(i)}$ plus a zero-mean Gaussian noise term $\tilde{\epsilon}^{t,(i)} \sim \mathcal{N}(\tilde{\epsilon}^t; 0, S^t)$,

$$\tilde{x}^{t,(i)} = B^t x^{t,(i)} + C^t y^t + d^t + \tilde{\epsilon}^{t,(i)}. \tag{22}$$

Using that $x^{t,(i)}$ in a similar fashion can be obtained as $x^{t,(i)} = \mu^t + \omega^{t,(i)}$, where $\omega^{t,(i)} \sim \mathcal{N}(\omega^t; 0, Q^t)$, we can rewrite Eq. (22) as

$$\tilde{x}^{t,(i)} = B^t\mu^t + C^t y^t + d^t + B^t\omega^{t,(i)} + \tilde{\epsilon}^{t,(i)}.$$

Given (θ^t, y^t), the stochastic components on the right hand side of this equation are $\omega^{t,(i)}$ and $\tilde{\epsilon}^{t,(i)}$ which are independent and Gaussian. Thereby, since $\tilde{x}^{t,(i)}$ is a linear combination of $\omega^{t,(i)}$ and $\tilde{\epsilon}^{t,(i)}$, we find that $\tilde{x}^{t,(i)}$ given (θ^t, y^t) is distributed according to a Gaussian distribution $\mathcal{N}(\tilde{x}^{t,(i)}; \tilde{\mu}^t, \tilde{Q}^t)$ with mean vector $\tilde{\mu}^t$ and covariance matrix $\tilde{Q}^t$ respectively given as

$$\tilde{\mu}^t = B^t\mu^t + C^t y^t + d^t \tag{23}$$

and

$$\tilde{Q}^t = B^t Q^t (B^t)^\top + S^t. \tag{24}$$

The requirement in Eq. (14) now states that the mean vector $\tilde{\mu}^t$ in Eq. (23) must be equal to μ^{*t} in Eq. (7) and that the covariance matrix $\tilde{Q}^t$ in Eq. (24) must be equal to Q^{*t} in Eq. (8). That is, we must have

$$B^t\mu^t + C^t y^t + d^t = \mu^t + K^t(y^t - H^t\mu^t) \tag{25}$$

and

$$B^t Q^t (B^t)^\top + S^t = (I_n - K^t H^t) Q^t. \tag{26}$$

Solving Eq. (25) with respect to $C^t y^t + d^t$ and inserting the result into Eq. (22), we obtain

$$\tilde{x}^{t,(i)} = B^t(x^{t,(i)} - \mu^t) + \mu^t + K^t(y^t - H^t \mu^t) + \tilde{\epsilon}^{t,(i)}. \tag{27}$$

Thereby, we see that in order to update $x^{t,(i)}$ we must specify appropriate B^t and S^t. To choose a procedure, one may either first choose S^t and thereafter compute B^t consistent with Eq. (26), or one may first choose B^t and then compute S^t consistent with Eq. (26). Below, we list three solutions that are particularly interesting.

Example 1. *By choosing all elements of B^t equal to zero, we obtain $\tilde{x}^{t,(i)}$ independent of $x^{t,(i)}$,*

$$\tilde{x}^{t,(i)} = \mu^t + K^t(y^t - H^t \mu^t) + \tilde{\epsilon}^{t,(i)}.$$

We then have $S^t = (I_n - K^t H^t) Q^t$, and $q(\tilde{x}^{t,(i)} | x^{t,(i)}, \theta^t, y^t)$ is simply equal to the assumed posterior model $f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t)$, i.e. the Gaussian distribution with mean and covariance given by Eqs. (7) *and* (8)*, respectively.*

Example 2. *By choosing all elements of S^t equal to zero, the update of $x^{t,(i)}$ becomes deterministic and equivalent to a square root EnKF. Specifically, Eq.* (27) *becomes equal to Eq.* (10)*, and Eq.* (26) *becomes equal to Eq.* (11)*. The distribution $q(\tilde{x}^{t,(i)} | x^{t,(i)}, \theta^t, y^t)$ is then a degenerate Gaussian distribution, or a delta function.*

Example 3. *By choosing*

$$B^t = I_n - K^t H^t \tag{28}$$

and

$$S^t = (I_n - K^t H^t) Q^t (K^t H^t)^\top \tag{29}$$

the update in Eq. (27) *becomes equivalent to the stochastic EnKF update in Eq.* (9)*. This result is proved in Appendix A.*

4.3 The optimal solution

The optimality criterion we consider for this situation is to minimise the expected value of the Mahalanobis distance $g(x^{t,(i)}, \tilde{x}^{t,(i)})$ in Eq. (16) for a general positive definite matrix Σ. The minimisation is to be solved with respect to B^t

and S^t under the restriction in Eq. (26) and, since S^t is a covariance matrix, the additional restriction that S^t is positive semidefinite.

To compute the optimal solution with respect to these criteria, we start out using that Σ^{-1} can be factorised as $\Sigma^{-1} = A^\top A$, $A \in \mathbb{R}^{n\times n}$. Hence, the function to be minimised, with respect to B^t and S^t, is

$$\mathrm{E}\big[g(x^{t,(i)}, \tilde{x}^{t,(i)})\big] = \mathrm{E}\Big[\big(A(\tilde{x}^{t,(i)} - x^{t,(i)})\big)^\top \big(A(\tilde{x}^{t,(i)} - x^{t,(i)})\big)\Big], \tag{30}$$

where the expectation is taken over the joint distribution $f_{x^t|\theta^t}(x^t|\theta^t)q(\tilde{x}^{t,(i)}|x^t, \theta^t, y^t)$. Using Eq. (27), we can write $A(\tilde{x}^{t,(i)} - x^{t,(i)})$ as

$$A(\tilde{x}^{t,(i)} - x^{t,(i)}) = A\left((B^t - I_n)(x^{t,(i)} - \mu^t) + K^t(y^t - H^t\mu^t) + \tilde{\epsilon}^{t,(i)}\right). \tag{31}$$

Since θ^t and y^t are treated as constants, the only stochastic components on the right hand side of Eq. (31) are x^t and $\tilde{\epsilon}^{t,(i)}$, which are independent and Gaussian. Thereby, $A(\tilde{x}^{t,(i)} - x^{t,(i)})$ is Gaussian since it is a linear combination of independent Gaussian variables. Moreover, from Eq. (31) we see that

$$\mathrm{E}\big[A(\tilde{x}^{t,(i)} - x^{t,(i)})\big] = AK^t(y^t - H^t\mu^t)$$

and

$$\mathrm{Cov}\big[A(\tilde{x}^{t,(i)} - x^{t,(i)})\big] = A(B^t - I_n)Q^t(B^t - I_n)^\top A^\top + AS^tA^\top.$$

Using that for any stochastic vector w we have $\mathrm{E}\big[w^\top w\big] = \mathrm{tr}[\mathrm{Cov}(w)] + \mathrm{E}[w]^\top \mathrm{E}[w]$, we can write Eq. (30) as

$$\begin{aligned}\mathrm{E}\Big[\big(A(\tilde{x}^{t,(i)} - x^{t,(i)})\big)^\top \big(A(\tilde{x}^{t,(i)} - x^{t,(i)})\big)\Big] &= \mathrm{tr}\big(A(B^t - I_n)Q^t(B^t - I_n)^\top A^\top\big) \\ &+ \mathrm{tr}\big(AS^tA^\top\big) + \big(AK^t(y^t - H^t\mu^t)\big)^\top \big(AK^t(y^t - H^t\mu^t)\big).\end{aligned} \tag{32}$$

We see that the last term in this equation is constant as a function of B^t and S^t. Thereby, to minimise Eq. (30) with respect to B^t and S^t we only need to minimise the sum of the two traces in Eq. (32). According to the restriction in Eq. (26) we must have

$$S^t = (I_n - K^tH^t)Q^t - B^tQ^t(B^t)^\top. \tag{33}$$

Using Eq. (33), we can write the sum of the two traces in Eq. (32) as a function of B^t only,

$$\begin{aligned}&\operatorname{tr}\left\{A(B^t - I_n)Q^t(B^t - I_n)^\top A^\top\right\} + \operatorname{tr}\left\{AS^tA^\top\right\}\\&\qquad\qquad = \operatorname{tr}\left\{-2AB^tQ^tA^\top + 2AQ^tA^\top - AK^tH^tQ^tA^\top\right\}.\end{aligned}$$

Here, only the first term is a function of B^t. Hence, minimising Eq. (30) with respect to B^t is equivalent to maximising

$$c(B^t) = \operatorname{tr}\left\{AB^tQ^tA^\top\right\} \tag{34}$$

with respect to B^t under the restriction that the matrix S^t in Eq. (33) is positive semidefinite.

To solve the optimisation problem stated above, we first rephrase it to a standardised form. To do so, we start with singular value decompositions of the two covariance matrices Q^t and $(I_n - K^tH^t)Q^t$,

$$Q^t = VDV^\top, \tag{35}$$

$$(I_n - K^tH^t)Q^t = U\Lambda U^\top, \tag{36}$$

where $U, V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, i.e. $UU^\top = U^\top U = I$ and $VV^\top = V^\top V = I_n$, and $D, \Lambda \in \mathbb{R}^{n\times n}$ are diagonal matrices. Inserting Eqs. (35) and (36) into Eq. (33) and defining

$$\tilde{S}^t = \Lambda^{-\frac{1}{2}}U^\top S^t U\Lambda^{-\frac{1}{2}}$$

and

$$\tilde{B}^t = \left(\Lambda^{-\frac{1}{2}}U^\top B^t V D^{\frac{1}{2}}\right)^\top$$

we get that Eq. (33) is equivalent to

$$\tilde{S}^t = I_n - (\tilde{B}^t)^\top \tilde{B}^t \tag{37}$$

and the objective function $c(B^t)$ in Eq. (34) can be rephrased in terms of $\tilde{B}^t$ as

$$\begin{aligned}\tilde{c}(\tilde{B}^t) &= \operatorname{tr}\left\{AU\Lambda^{\frac{1}{2}}(\tilde{B}^t)^\top D^{\frac{1}{2}}V^\top A^\top\right\} = \operatorname{tr}\left\{\tilde{B}\Lambda^{\frac{1}{2}}U^\top A^\top AQ^tVD^{-\frac{1}{2}}\right\}\\&= \operatorname{tr}\left\{\tilde{B}^tZ^t\right\},\end{aligned} \tag{38}$$

where

$$Z^t = \Lambda^{\frac{1}{2}} U^\top A^\top A Q^t V D^{-\frac{1}{2}}. \tag{39}$$

Recognising that the matrix $\tilde{S}^t$ is positive semidefinite if and only if S^t is positive semidefinite, the rephrased optimisation problem is thereby to maximise $\tilde{c}(\tilde{B}^t)$ in Eq. (38) with respect to $\tilde{B}^t$ under the constraint that $\tilde{S}^t$ in Eq. (37) is positive semidefinite. To solve this standardised optimisation problem we can apply the following theorem for which a proof is given in Appendix B.

Theorem 1. *For a square matrix $Z \in \mathbb{R}^{n\times n}$ of full rank and with singular value decomposition $Z = PGF^\top$ the maximum value for $tr(\tilde{B}Z)$, $\tilde{B} \in \mathbb{R}^{n\times n}$ under the restriction that $\tilde{S} = I_n - \tilde{B}^\top \tilde{B}$ is positive semidefinite occurs only for*

$$\tilde{B} = FP^\top.$$

To apply Theorem 1, we first need to argue why the matrix Z^t in Eq. (39) has full rank. Since Q^t and $(I_n - K^t H^t)Q^t$ are positive definite matrices, D and Λ are invertible. Thereby also $D^{\frac{1}{2}}$ and $\Lambda^{\frac{1}{2}}$ are invertible. V and U are both orthogonal and thereby invertible. Finally, as we have required Σ to be positive definite, Σ is invertible, and when Σ is invertible, A is also invertible. Thereby, Z^t is given as a product of invertible matrices and is therefore itself invertible and has full rank.

According to Theorem 1 the solution to our optimisation problem in standardised form is $\tilde{B}^t = FP^\top$. We thereby get that

$$\tilde{S}^t = I_n - (FP^\top)^\top FP^\top = I_n - PF^\top FP^\top = 0,$$

i.e. all elements in $\tilde{S}^t$, and hence all elements in S^t, are zero. The solution to our optimisation problem thereby corresponds to a square root EnKF. The corresponding optimal value for B^t is

$$B^t = U\Lambda^{\frac{1}{2}} PF^\top D^{-\frac{1}{2}} V^\top.$$

4.4 Parameter simulation

Before we can construct $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$, we need to simulate a parameter $\theta^{t,(i)}|x^{t,-(i)}, y^t \sim f_{\theta^t|x^{t,-(i)},y^t}(\theta^t|x^{t,-(i)}, y^t)$. As explained in Section 3.4, this can be done with a Gibbs sampler. To construct the Gibbs sampler, we need to derive the full conditional distributions $f_{x^t|\theta^t,y^t}(x^t|\theta^t,y^t)$ and $f_{\theta^t|x^t,x^{t,-(i)}}(\theta^t|x^t,x^{t,-(i)})$, where now $\theta^t=(\mu^t,Q^t)$. The first distribution, $f_{x^t|\theta^t,y^t}(x^t|\theta^t,y^t)$, is already known and is a Gaussian with parameters μ^{*t} and Q^{*t} given by Eqs. (7) and (8), respectively. To derive the second distribution, $f_{\theta^t|x^t,x^{t,-(i)}}(\theta^t|x^t,x^{t,-(i)})$, we first factorise it as

$$f_{\theta^t|x^t,x^{t,-(i)}}(\theta^t|x^t,x^{t,-(i)}) = f_{Q^t|x^t,x^{t,-(i)}}(Q^t|x^t,x^{t,-(i)})f_{\mu^t|Q^t,x^t,x^{t,-(i)}}(\mu^t|Q^t,x^t,x^{t,-(i)}).$$

Since conjugate priors are chosen for μ^t and Q^t, and since x^t, $x^{t,(1)},\dots,x^{t,(M)}$ are independent and identically distributed given θ^t, it can be shown that $f_{Q^t|x^t,x^{t,-(i)}}(Q^t|x^t,x^{t,-(i)})$ is an inverse Wishart distribution,

$$f_{Q^t|x^t,x^{t,-(i)}}(Q^t|x^t,x^{t,-(i)}) = \mathcal{W}^{-1}(Q^t;\tilde{V},\tilde{\nu}),$$

where

$$\tilde{\nu} = \nu + M$$

and

$$\tilde{V} = V + C^{t,(i)} + \frac{\kappa M}{\kappa + M}\left(\bar{x}^{t,(i)} - \mu_0\right)\left(\bar{x}^{t,(i)} - \mu_0\right)^\top,$$

where

$$\bar{x}^{t,(i)} = \frac{1}{M}\left(x^t + \sum_{j\neq i} x^{t,(j)}\right)$$

and

$$C^{t,(i)} = \left(x^t - \bar{x}^{t,(i)}\right)\left(x^t - \bar{x}^{t,(i)}\right)^\top + \sum_{j\neq i}\left(x^{t,(j)} - \bar{x}^{t,(i)}\right)\left(x^{t,(j)} - \bar{x}^{t,(i)}\right)^\top,$$

and $f_{\mu^t|Q^t,x^t,x^{t,-(i)}}(\mu^t|Q^t,x^t,x^{t,-(i)})$ is a Gaussian distribution,

$$f_{\mu^t|Q^t,x^t,x^{t,-(i)}}(\mu^t|Q^t,x^t,x^{t,-(i)}) = \mathcal{N}(\mu^t;\tilde{\mu}_0,\tilde{\kappa}^{-1}Q^t),$$

where

$$\tilde{\mu}_0 = \frac{\kappa\mu_0 + M\bar{x}^{t,(i)}}{\kappa + M}$$

and

$$\tilde{\kappa} = \kappa + M.$$

5 First-order Markov chain assumed model

In this section, we describe how the general updating procedure described in Section 3 can be applied when the elements of the state vector x^t are categorical variables, $x_j^t \in \{0, 1, \ldots, K-1\}$, and x^t is restricted to have a one-dimensional spatial arrangement. As in Loe and Tjelmeland (2021), we then propose to let $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{y^t|x^t}(y^t|x^t)$ constitute an HMM.

5.1 Specification of the assumed model

Suppose $x^t = (x_1^t, \ldots, x_n^t)$ is a vector of n categorical variables, $x_j^t \in \{0, 1, \ldots, K-1\}$, and suppose x^t has a spatial arrangement along a line. A natural choice of model for $f_{x^t|\theta^t}(x^t|\theta^t)$ is then a first-order Markov chain,

$$f_{x^t|\theta^t}(x^t|\theta^t) = f(x_1^t|\theta^t) \prod_{j=2}^{n} f(x_j^t|x_{j-1}^t, \theta^t). \tag{40}$$

Moreover, suppose $y^t = (y_1^t, \ldots, y_n^t)$ is a vector of n variables, $y_j^t \in \mathbb{R}$, so that we have one observation y_j^t for each component x_j^t of x^t, and assume that the y_j^t's are conditionally independent given x^t, with y_j^t only dependent on x_j^t,

$$f_{y^t|x^t}(y^t|x^t) = \prod_{j=1}^{n} f_{y_j^t|x_j^t}(y_j^t|x_j^t).$$

Given θ^t, the models $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{y^t|x^t}(y^t|x^t)$ constitute an HMM. The corresponding posterior model $f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t)$ is then also a first-order Markov chain whose initial and transition probabilities can be computed with the the forward-backward algorithm for HMMs (e.g., Künsch, 2000).

The parameter θ^t may in this context represent the initial and transition probabilities of the assumed first-order Markov chain $f_{x^t|\theta^t}(x^t|\theta^t)$. In the following, we let

$$\theta^t = \left(\{\theta_1^t(i)\}_{i=0}^{K-1}, \{\theta_2^{t,k}(i)\}_{i,k=0}^{K-1}, \ldots, \{\theta_n^{t,k}(i)\}_{i,k=0}^{K-1} \right),$$

where $\theta_1(i)^t, \theta_j^{t,k}(i) \in (0,1)$, $\sum_{i=0}^{K-1} \theta_j^{t,k}(i) = 1$, and

$$f(x_1^t = i|\theta^t) = \theta_1^t(i)$$

and

$$f(x_j^t = i | x_{j-1}^t = k, \theta^t) = \theta_j^{t,k}(i),$$

for $i, k = 0, \ldots, K-1$ and $j = 2, \ldots, n$. For convenience, we also define

$$\theta_1^t = (\theta_1^t(0), \theta_1^t(1), \ldots, \theta_1^t(K-1))$$

and

$$\theta_j^{t,k} = (\theta_j^{t,k}(0), \theta_j^{t,k}(1), \ldots, \theta_j^{t,k}(K-1)).$$

As recommended in Section 3.4, we choose $f_{\theta^t}(\theta^t)$ as a conjugate prior for $f_{x^t|\theta^t}(x^t|\theta^t)$. Specifically, we assume that all the vectors θ_1^t, $\theta_2^{t,0}, \ldots,$ $\theta_2^{t,K-1}$, $\theta_3^{t,0}, \ldots,$ $\theta_3^{t,K-1}, \ldots, \theta_n^{t,0}, \ldots, \theta_n^{t,K-1}$ are a priori independent, so that

$$f_{\theta^t}(\theta^t) = f_{\theta_1^t}(\theta_1^t) \prod_{j,k} f_{\theta_j^{t,k}}(\theta_j^{t,k}).$$

Moreover, we choose $f_{\theta_1^t}(\theta_1^t)$ as a Dirichlet distribution with (known) hyperparameters $\alpha_1^t(0), \ldots, \alpha_1^t(K-1)$,

$$f_{\theta_1^t}(\theta_1^t) \propto \prod_{i=0}^{K-1} \theta_1^t(i),$$

and choose each $f_{\theta_j^{t,k}}(\theta_j^{t,k})$ as a Dirichlet distribution with (known) hyperparameters $\alpha_j^{t,k}(0), \ldots, \alpha_j^{t,k}(K-1)$,

$$f_{\theta_j^{t,k}}(\theta_j^{t,k}) \propto \prod_{i=0}^{K-1} \theta_j^{t,k}(i)^{\alpha_j^{t,k}(i)}.$$

5.2 Class of updating distributions

Because of the discrete context, the criterion in Eq. (15) can be written as a sum, i.e.

$$f_{x^t|y^t,\theta^t}(\tilde{x}^{t,(i)}|y^t, \theta^t) = \sum_{x^{t,(i)} \in \Omega_x} f_{x^t|\theta^t}(x^{t,(i)}|\theta^t) q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t). \tag{41}$$

Brute force, the updating distribution $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$ now represents a transition matrix, and there are $K^n(K^n-1)$ transition probabilities that need to be specified. Even when n is only moderately large this is too computationally de-

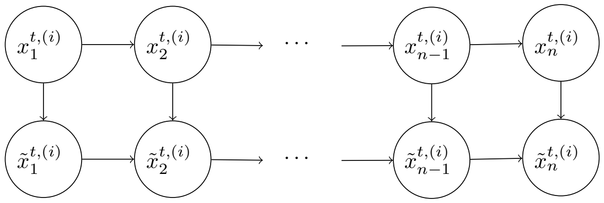


Figure 3: Graphical illustration of enforced dependencies between the variables in a prior sample $x^{t,(i)}$ and corresponding posterior sample $\tilde{x}^{t,(i)}$, given θ^t and y^t, when the assumed forecast model is chosen as a first-order Markov chain.

manding to cope with. To simplify the situation, we therefore enforce a certain dependency structure for $q(\tilde{x}^{t,(i)}|x^{t,(i)},\theta^t,y^t)$ as illustrated in Figure 3. We can then factorise $q(\tilde{x}^{t,(i)}|x^{t,(i)},\theta^t,y^t)$ as

$$q(\tilde{x}^{t,(i)}|x^{t,(i)},\theta^t,y^t) = q(\tilde{x}_1^{t,(i)}|x_1^{t,(i)},\theta^t,y^t)\prod_{j=2}^{n} q(\tilde{x}_j^{t,(i)}|\tilde{x}_{j-1}^{t,(i)},x_j^{t,(i)},\theta^t,y^t). \tag{42}$$

The number of quantities required to specify $q(\tilde{x}^{t,(i)}|x^{t,(i)},\theta^t,y^t)$ thereby reduces to $K(K-1)+(n-1)K^2(K-1)$, or more specifically $K(K-1)$ quantities for $q(\tilde{x}_1^{t,(i)}|x_1^{t,(i)},\theta^t,y^t)$ and $K^2(K-1)$ quantities for each factor $q(\tilde{x}_j^{t,(i)}|\tilde{x}_{j-1}^{t,(i)},x_j^{t,(i)},\theta^t,y^t)$, $j=2,\ldots,n$. As this is a linear, rather than an exponential, function of n, n can be large without causing trouble.

According to the requirement in Eq. (41), $q(\tilde{x}^{t,(i)}|x^{t,(i)},\theta^t,y^t)$ must be constructed such that marginalising out $x^{t,(i)}$ from the joint distribution $f_{x^t|\theta^t}(x^{t,(i)}|\theta^t)q(\tilde{x}^{t,(i)}|x^{t,(i)},\theta^t,y^t)$ returns the posterior Markov chain model $f_{x^t|\theta^t,y^t}(\tilde{x}^{t,(i)}|\theta^t,y^t)$. However, the problem of constructing such a $q(\tilde{x}^{t,(i)}|x^{t,(i)},\theta^t,y^t)$, different from $f_{x^t|\theta^t,y^t}(x^t|\theta^t,y^t)$ itself, is generally too intricate to solve. Therefore, we need an approximate approach. As in Loe and Tjelmeland (2021), we propose to replace the requirement of retaining the whole Markov chain model $f_{x^t|\theta^t,y^t}(x^t|\theta^t,y^t)$ with the requirement that only the bivariate probabilities $f_{x_j^t,x_{j+1}^t|\theta^t,y^t}(x_j^t,x_{j+1}^t|\theta^t,y^t)$ are retained, i.e.

$$f_{\tilde{x}_j^{t,(i)},\tilde{x}_{j+1}^{t,(i)}|\theta^t,y^t}(x_j^t,x_{j+1}^t|\theta^t,y^t) = f_{x_j^t,x_{j+1}^t|\theta^t,y^t}(x_j^t,x_{j+1}^t|\theta^t,y^t), \tag{43}$$

for $j=1,\ldots,n-1$. This means that, under the assumption that the assumed model is correct, the distribution of the updated sample $\tilde{x}^{t,(i)}$ given (θ^t,y^t) is not equal to the first-order Markov chain $f_{x^t|\theta^t,y^t}(x^t|\theta^t,y^t)$, but that each pair

$(\tilde{x}_j^{t,(i)}, \tilde{x}_{j+1}^{t,(i)}), j = 1, \ldots, n-1$, is marginally distributed according to the bivariate distribution $f_{x_j^t, x_{j+1}^t|\theta^t, y^t}(x_j^t, x_{j+1}^t|\theta^t, y^t)$ of the Markov chain.

5.3 The optimal solution

The optimality criterion we consider for this situation is to minimise the expected number of components of $x^{t,(i)}$ that are different from their corresponding components in $\tilde{x}^{t,(i)}$; that is, we want to minimise the expected value of the function $g(x^{t,(i)}, \tilde{x}^{t,(i)})$ in Eq. (18). Minimising $\mathrm{E}\left[g(x^{t,(i)}, \tilde{x}^{t,(i)})\right]$ is then equivalent to maximising

$$\mathrm{E}\left[\sum_{j=1}^{n} 1(x_j^{t,(i)} = \tilde{x}_j^{t,(i)})\right] \tag{44}$$

where the expectation is taken over $f_{x^t|\theta^t}(x^t|\theta^t)q(\tilde{x}^{t,(i)}|x^t, \theta^t, y^t)$. We are thereby faced with a constrained optimisation problem where we want to maximise, with respect to $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$, the function in Eq. (44) under the condition in Eq. (43) and under the condition that $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$ can be factorised as in Eq. (42).

Loe and Tjelmeland (2021) propose a dynamic programming algorithm for solving the optimisation problem stated above when x_j^t is binary, $x_j^t \in \{0, 1\}$. The proposed algorithm is based on that the maximum value of Eq. (44) can be computed recursively since

$$\begin{aligned}\max_{q_{k:n}^t} \mathrm{E}\left[\sum_{j=k}^{n} 1(x_j^{t,(i)} = \tilde{x}_j^{t,(i)})\right] &= \max_{q_{k:n}^t} \mathrm{E}\left[1(x_k^{t,(i)} = \tilde{x}_k^{t,(i)}) + \sum_{j=k+1}^{n} 1(x_j^{t,(i)} = \tilde{x}_j^{t,(i)})\right] \\ &= \max_{q_k^t} \mathrm{E}\left[1(x_k^{t,(i)} = \tilde{x}_k^{t,(i)}) + \max_{q_{k+1:n}^t} \mathrm{E}\left[\sum_{j=k+1}^{n} 1(x_j^{t,(i)} = \tilde{x}_j^{t,(i)})\right]\right]\end{aligned} \tag{45}$$

where $q_k^t = q(\tilde{x}_k^{t,(i)}|\tilde{x}_{k-1}^{t,(i)}, x_k^{t,(i)}, \theta^t, y^t)$, $q_1^t = q(\tilde{x}_1^{t,(i)}|x_1^{t,(i)}, \theta^t, y^t)$, and $q_{k:n}^t = (q_k^t, \ldots, q_n^t)$. The algorithm starts with a 'backward' recursion where, for $k = n, n-1, \ldots, 1$, Eq. (45) and the optimal value of q_k^t are computed as functions of $q_{1:k-1}^t = (q_1^t, \ldots, q_{k-1}^t)$. At the final step of the backward recursion the whole expectation in Eq. (44) is thereby computed, along with the optimal value for q_1^t. The algorithm then proceeds with a 'forward' recursion where, for $k = 2, \ldots, n$, we recursively compute the optimal values for $q_2^t, \ldots, q_n^t$.

5.4 Parameter simulation

To construct the Gibbs sampler described in Section 3.4, we need to be able to simulate from the distributions $f_{x^t|\theta^t,y^t}(x^t|\theta^t,y^t)$ and $f_{\theta^t|x^t,x^{t,-(i)}}(\theta^t|x^t,x^{t,-(i)})$. From Section 5.1 we know that $f_{x^t|\theta^t,y^t}(x^t|\theta^t,y^t)$ now is a first-order Markov chain with transition probabilities that are easy to compute. When it comes to $f_{\theta^t|x^t,x^{t,-(i)}}(\theta^t|x^t,x^{t,-(i)})$, it can easily be shown that $\theta_1^t|x^t,x^{t,-(i)}$ is Dirichlet distributed with parameters

$$\tilde{\alpha}_1^t(r) = \alpha_1^t(r) + 1(x_1^t = r) + \sum_{m \neq i} 1\left(x_1^{t,(m)} = r\right),$$

for $r = 0, \ldots, K-1$. Similarly, it can be shown that each $\theta_j^{t,k}|x^t,x^{t,-(i)}$ is Dirichlet distributed with parameters

$$\tilde{\alpha}_j^{t,k}(r) = \alpha_j^{t,k}(r) + 1(x_{j-1} = k, x_j = r) + \sum_{m \neq i} 1\left(x_{j-1}^{t,(m)} = k, x_j^{t,(m)} = r\right)$$

for $r = 0, \ldots, K-1$. Moreover, all the parameters are independent a posteriori,

$$f_{\theta^t|x^t,x^{t,-(i)}}(\theta^t|x^t,x^{t,-(i)}) = f_{\theta_1^t|x^t,x^{t,-(i)}}(\theta_1^t|x^t,x^{t,-(i)}) \prod_{j,k} f_{\theta_j^{t,k}|x^t,x^{t,-(i)}}(\theta_j^{t,k}|x^t,x^{t,-(i)}).$$

6 Simulation example with a linear-Gaussian assumed model

In this section, we present a simulation example for the situation described in Section 4. The example is based on an experimental setup previously used in Myrseth and Omre (2010). In the following, we first describe how we generate a reference time series and simulate corresponding observations. Thereafter, we specify the precise assumed model we are using, and finally we present and discuss simulation results.

6.1 Experimental setup

To generate a reference time series $\{x^t\}_{t=1}^T$ that we consider as the true unobserved state process we adopt the same setup as in Myrseth and Omre (2010). At each time t, we assume that the state vector $x^t = (x_1^t, \ldots, x_n^t)$ consists of $n = 100$ continuous variables so that $\Omega_x = \mathbb{R}^{100}$. The latent process is defined from time

1 to time $T = 11$. The values of the initial state vector, x^1, is generated from a Gaussian distribution with zero mean, where the variance of each component is 20 and where the correlation between elements r and s in x^1 is

$$c(r,s) = \exp\left\{-\frac{3|r-s|}{20}\right\}.$$

Myrseth and Omre (2010) define two deterministic ways to generate $x^t, t = 2,\ldots,T$ from x^1, one linear forward function and one non-linear. We adopt the same linear forward function as used there, but not the same non-linear function. The non-linear forward function used in Myrseth and Omre (2010) induces a light-tailed bi-modal marginal distribution for each component in the state vector at time $t = T$. We construct instead a forward function which produces a heavy-tailed one-mode marginal distribution for time $t > 1$.

For $t = 2,\ldots,T$, the linear forward function we use is defined by

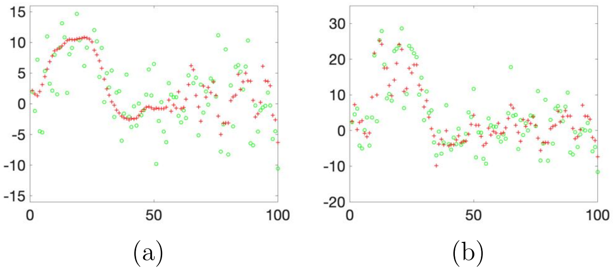
$$x^t = \xi^{t-1}x^{t-1},$$

where ξ^{t-1} is an $n \times n$ matrix defined so that for $j = 5t-4,\ldots,5t+5$, element j in x^t is set equal to the average of elements $\max\{1, j-4\}$ to $j+5$ in x^{t-1}, whereas the remaining elements in x^t equal the corresponding elements in x^{t-1}. The effect of this forward function is that the first part of the vector x^t is a smoothed version of the first part of x^1, whereas the rest of x^t equals the corresponding part of x^1. When the time t increases, the part that has been smoothed also increases.

For the non-linear forward function, we simply transform the Gaussian distributed elements in the state vector at time $t = 1$ to be from a (scaled) t-distribution at any later time $t > 1$. More specifically, element j in x^2 is defined from the corresponding element in x^1 by

$$x_j^2 = \sqrt{20}F_{\mathcal{T}}^{-1}\left(\Phi\left(\frac{x_j^1}{\sqrt{20}}\right), 100\right), \tag{46}$$

where $F_{\mathcal{T}}(\cdot,\nu)$ and $\Phi(\cdot)$ are the cumulative distribution functions for a t-distribution with ν degrees of freedom and a standard normal distribution, respectively. Thus, the marginal distribution of each element in x^2 is a t-distribution with 100 degrees of freedom. For later times $t > 2$, each element j in x^t is defined



(a)

(b)

Figure 4: The reference state vector (red crosses) at time $t = T$ for the (a) linear and (b) non-linear forward model cases, and the simulated observations (green circles) at the same time. Note that a few of the observations are outside the range of the vertical axis.

from the corresponding element in x^{t-1} by

$$x_j^t = \sqrt{20} F_{\mathcal{T}}^{-1}\left(F_{\mathcal{T}}\left(\frac{x_j^{t-1}}{\sqrt{20}}, \nu_{t-1}\right), \nu_t\right), \tag{47}$$

where $\nu_t = 100/(2t-3)$. Thus, the marginal distribution for each element gets heavier and heavier tails when the time t increases.

Having generated a reference time series $\{x^t\}_{t=1}^T$ as described above, observations are simulated for each time $t = 1, \ldots, T$. For each time $t = 1, \ldots, T$ an observation vector y^t is simulated according to

$$y^t | x^t \sim \mathcal{N}\left(y^t; x^t, 20 I_n\right). \tag{48}$$

The reference state vectors for the linear and the non-linear models at time $t = T$ and the corresponding simulated observations at that time step are shown in Figure 4.

6.2 Details of the assumed model

The assumed model is as specified in Section 4.1. The hyper-prior in Eqs. (19) and (20) for $\theta^t = (\mu^t, Q^t)$ is specified by four hyper-parameters: μ_0, κ, ν and V. We choose values for these hyper-parameters to get a vague, but proper prior for θ^t, and use the same values for all time steps. We set all the elements of $\mu_0 \in \mathbb{R}^n$ equal to zero, and set $\kappa = 10$, $\nu = n + 1.1$ and $V = (\nu - n - 1) I_n$. Note that this in particular gives $\mathrm{E}[Q^t] = I_n$ a priori. For the likelihood $f_{y^t|x^t}(y^t|x^t)$ we use the same distribution as the one we used to simulate the data, i.e. $f_{y^t|x^t}(y^t|x^t)$ is specified by Eq. (48).

6.3 Simulation results

When evaluating the performance of the proposed approach, the results are compared with several other variants of EnKF. When updating one of the ensemble members, there are two important steps. The first step is how to generate or estimate μ^t and Q^t based on the prediction ensemble. The second step is how to use these μ^t and Q^t values to update the ensemble member in question. We consider tree variants of the first step. The first is what we propose in this report, to sample μ^t and Q^t from a posterior distribution given the new observation y^t and all ensemble members, except the member which is to be updated. For the function $g(x^{t,(i)}, \tilde{x}^{t,(i)})$ we here use the Eucledian distance, i.e. $\Sigma = I_n$. The second is what Myrseth and Omre (2010) are advocating, to sample μ^t and Q^t from a posterior distribution given all the ensemble members, including also the member that is going to be updated, but not given the new observation y^t. The third is the standard procedure in EnKF, to estimate μ^t and Q^t based on all the ensemble members. For how to update an ensemble member when values of μ^t and Q^t are given, we consider two variants. The first is the square-root filter we found to be optimal in Section 4.3 and the second is the standard stochastic EnKF update procedure specified in Eq. (9). By combining each of the three variants of how to generate μ^t and Q^t with each of the two variants of how to update the ensemble members, one can define six updating procedures. We present results for all the six combinations.

Using the linear forward model described in Section 6.1, the prediction ensembles at time $T = 11$ in one run of each of the six procedures considered, with $M = 19$ ensemble members, are shown in Figure 5. The ensemble members, drawn with solid lines in the figure, should thus be considered as (approximate) samples from the distribution $p_{x^{11}|y^{1:10}}(x^{11}|y^{1:10})$. For comparison, the latent true state vector at time $T = 11$ is also shown, with red crosses. The upper, middle and lower lines show results when using our proposed procedure for generating μ^t and Q^t, when using the procedure in Myrseth and Omre (2010) for the same, and when using empirical estimates, respectively. The left and right columns show results when using our optimal square-root filter to update the ensemble members, and when using the standard stochastic EnKF update, respectively.

The most striking difference between the six cases is the spread of the ensemble

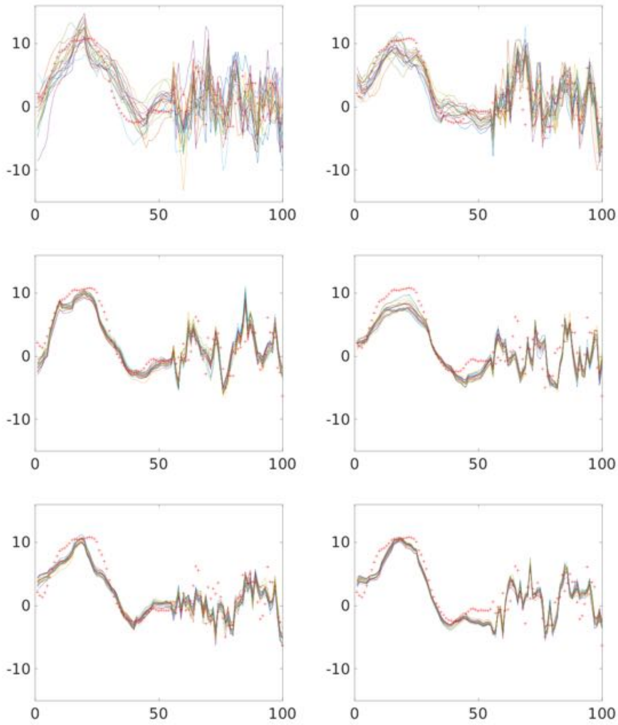


Figure 5: Gaussian linear example: Prediction ensemble at time $T = 11$ when using $M = 19$ ensemble members. The upper, middle and lower rows are when using our proposed procedure for generating μ^t and Q^t, when using the procedure of Myrseth and Omre (2010) for the same, and when using empirical estimates, respectively. The left and right columns are when updating with our optimal square-root filter and when using the standard stochastic EnKF procedure, respectively. The ensemble members are shown with solid lines and the latent true state is shown with red crosses.

members. In the four lower figures in Figure 5, the spread is very small, and as a result the latent true value is in most places outside the spread of the ensemble. For the standard stochastic EnKF procedure, shown in the lower right figure, this should come as no surprise as it is well known that this procedure tends to underestimate the uncertainty. What is more surprising is that the increase of the spread is so small when instead using the procedure proposed in Myrseth and Omre (2010), shown in the middle right figure. The difference in the spread of the ensemble members in each of the figures in the middle row and the corresponding figure in the upper row is also striking, when remembering the very small difference in the procedures used to generate the figures. The only difference between the procedures is what to condition on when generating values for μ^t and Q^t. In the procedures used to generate the figures in the middle row one is conditioning on all the ensemble members, but not the new data. In the procedure for the upper row one is conditioning on the new data and all the ensemble members except the ensemble member that is to be updated. Other simulation runs not included in this report show that most of the difference in the results comes from not conditioning on the ensemble member that is to be updated. The effect of including the new data in the conditioning set is clearly visible, but still small compared to the effect of not conditioning on the ensemble member that is to be updated.

In the four lower plots in Figure 5 the latent true state vector is in most positions outside the spread of the ensemble members. As such, these ensemble members do not give a realistic representation of our information about x^{11}. In the two upper plots in the same figure, the latent true state is in most positions inside the spread of the ensemble members. These ensembles may therefore give a better representation of the uncertainty. However, the spread in the ensemble members is larger in the upper left plot than in the upper right plot. So an interesting question is therefore which of the two that gives the best representation of our information about x^{11}. It is of course not necessarily the procedure that gives the largest spread that gives the best representation of uncertainty. To provide one answer to this question, one can first observe that in a perfect model, the variables $x^{t,(1)}, \ldots, x^{t,(M)}, x^t$ are exchangeable. One way to measure to what degree the spread of the ensemble members gives a realistic representation of the

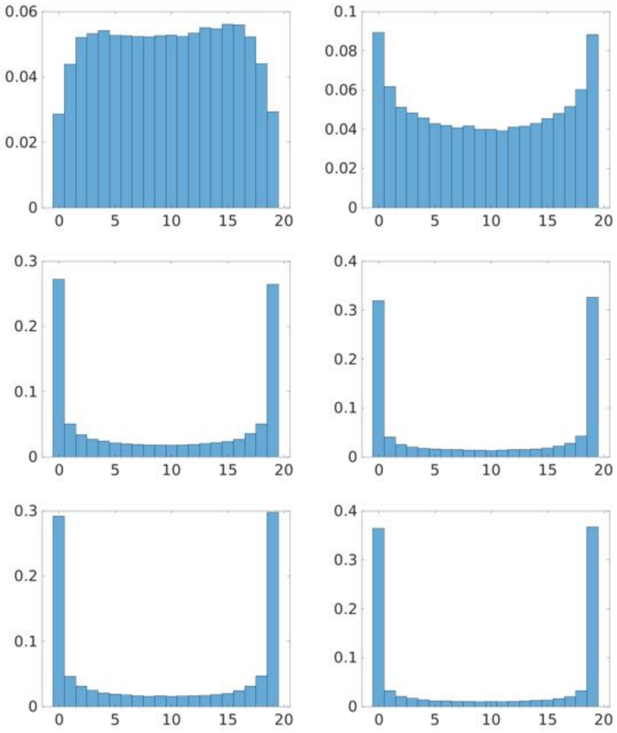


Figure 6: Gaussian linear example: Estimated distribution for Z in Eq. (49) when using $M = 19$ ensemble members. The upper, middle and lower rows are when using our proposed procedure for generating μ^t and Q^t, when using the procedure of Myrseth and Omre (2010) for the same, and when using empirical estimates, respectively. The left and right columns are when updating with our optimal square-root filter and when using the standard stochastic EnKF procedure, respectively.

uncertainty is therefore to study the distribution of

$$Z = \sum_{i=1}^{M} 1(x_j^{t,(i)} \leq x_j^t), \tag{49}$$

where the index j is sampled uniformly on the integers from 1 to n. In the perfect model Z has a uniform distribution on the integers zero to M. Repeating the simulation procedures leading to the plots in Figure 5 one thousand times, randomising also over the latent state vector, the plots in Figure 6 show the estimated distributions for Z for each of the six filtering procedures. The four lower plots in this figure just confirm what we saw in Figure 5, the latent state value is very often more extreme than all the ensemble members. The distributions in the two upper plots are neither perfectly uniform, but we see that the distribution in the upper left plot is slightly closer to being uniform than the upper right one. We thereby conclude that of the six procedures tried here, it is our proposed procedure that best represents our knowledge about x^{11}.

Above, we presented simulation experiments for the six ensemble updating procedures we have defined, for a linear forward model and with $M = 19$ ensemble members. We have also done similar simulation experiments for both smaller and larger ensemble sizes M, and for the non-linear forward function discussed in Section 6.1. There are two main lessons to learn from these experiments. The first is that the differences between the six methods gradually reduce when the number of ensemble members increases, and for M large enough they all behave essentially the same. It should, however, be remembered that in typical applications of the EnKF, the dimension of the state vector, n, is much larger than the number of ensemble members, M. As one example, the plots in Figure 7 are the same type of plots as in Figure 6, but for runs with $M = 199$ ensemble members.

The second lesson we learn from the simulation experiments, is that the results when using our non-linear forward function is quite similar to what we have for the linear forward function. As one example, Figures 8 and 9 show similar plots as in Figures 5 and 6, but for the non-linear forward function defined by Eqs. (46) and (47). Again we see that the upper left plot in Figure 9 is the one closest to being uniform. Also when using the non-linear forward function the differences between the six methods gradually vanish when the number of ensemble members, M, increases. Of course, that the results for our non-linear forward function are similar to the results for the linear function, does not imply that this is generally true for all non-linear forward functions. We have for example not studied how the various procedures perform with a forward function inducing skewed distributions for the state vector.

7 Simulation example with a Markov chain assumed model

In this section, we demonstrate the proposed updating procedure in a simulation example where the state vector consists of binary variables and $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{y^t|x^t}(y^t|x^t)$ constitute an HMM as described in Section 5. We first describe

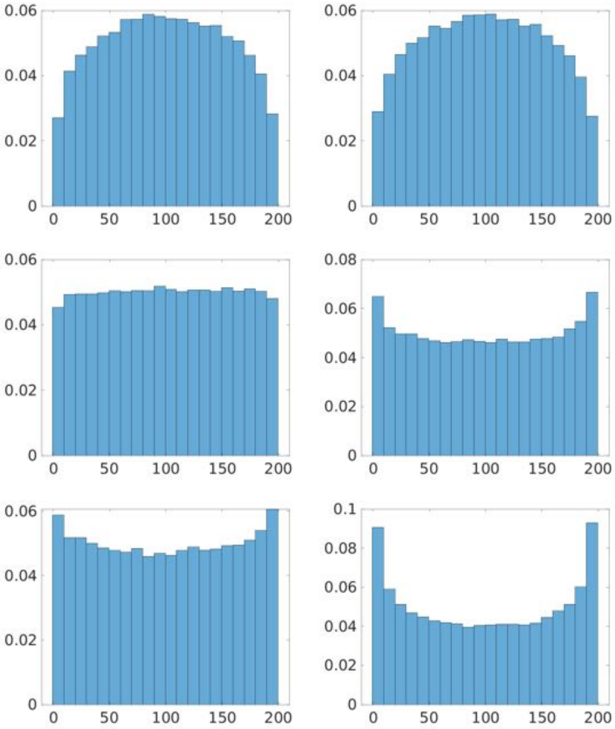


Figure 7: Gaussian linear example: Estimated distribution for Z when using $M = 199$ ensemble members. The upper, middle and lower rows are when using our proposed procedure for generating μ^t and Q^t, when using the procedure of Myrseth and Omre (2010) for the same, and when using empirical estimates, respectively. The left and right columns are when updating with our optimal square-root filter and when using the standard stochastic EnKF procedure, respectively.

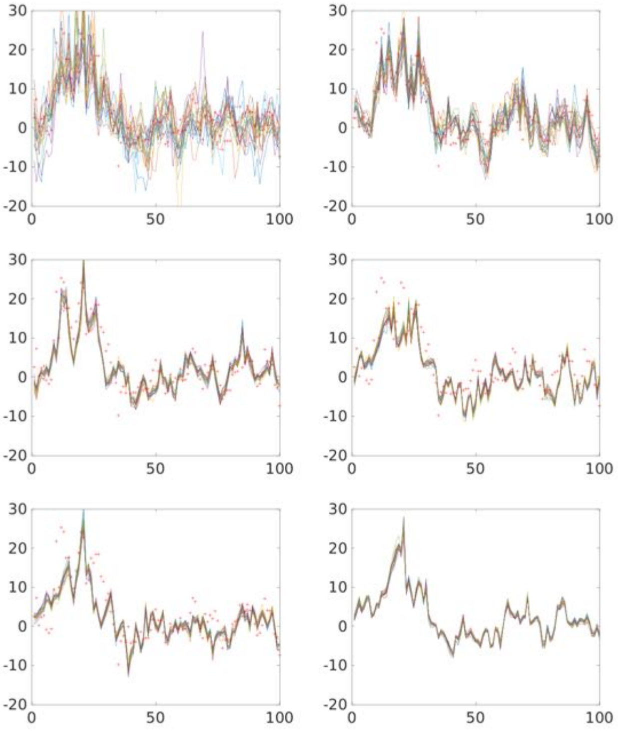


Figure 8: Gaussian non-linear example: Prediction ensemble at time $T = 11$ when using $M = 19$ ensemble members. The upper, middle and lower rows are when using our proposed procedure for generating μ^t and Q^t, when using the procedure of Myrseth and Omre (2010) for the same, and when using empirical estimates, respectively. The left and right columns are when updating with our optimal square-root filter and when using the standard stochastic EnKF procedure, respectively. The ensemble members are shown with solid lines and the latent true state is shown with red crosses.

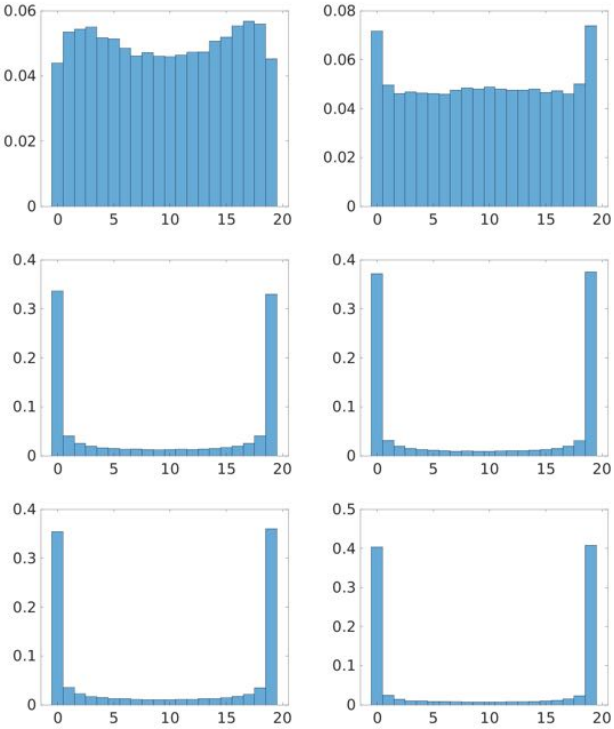


Figure 9: Gaussian non-linear example: Estimated distribution for Z when using $M = 19$ ensemble members. The upper, middle and lower rows are when using our proposed procedure for generating μ^t and Q^t, when using the procedure of Myrseth and Omre (2010) for the same, and when using empirical estimates, respectively. The left and right columns are when updating with our optimal square-root filter and when using the standard stochastic EnKF procedure, respectively.

the experimental setup of the simulation example in Section 7.1, and thereafter we present and discuss the simulation results in Section 7.2.

7.1 Experimental setup

The simulation example involves a state process $\{x^t\}_{t=1}^T$ with $T = 100$ time steps, and the state vector x^t at each time step is a vector of $n = 400$ binary variables, $x_j^t \in \{0, 1\}$. The initial distribution $p_{x^1}(x^1)$ and the forward model $p_{x^t|x^{t-1}}(x^t|x^{t-1})$ of the unobserved x^t-process are the same as in the simulation example of Loe and Tjelmeland (2021). For simplicity, we do not discuss the technical details of this model here, but one should note that the generated state vector x^t at any time t is not a first-order Markov chain. Moreover, the process is inspired by how water comes through to an oil-producing well in a petroleum reservoir. In this context, we let the t in x_j^t represent time and j the location in the well, and the values zero and one represent oil and water, respectively. Hence, the event $x_j^t = 0$ indicates the presence of oil in location j at time t, while the event $x_j^t = 1$ indicates the presence of water.

An image of a state process $\{x^t\}_{t=1}^T$ generated using the true model specified above is shown in Figure 10(a), where the colours black and white represent the values zero (oil) and one (water), respectively. Based on this reference state process, a corresponding observation process $\{y^t\}_{t=1}^T$ is generated by simulating, independently for each time step $t = 1, \dots, T$ and for each node $j = 1, \dots, n$, an observation y_j^t from a Gaussian distribution with mean x_j^t and variance $\sigma^2 = 2^2$. Figure 10(b) shows a grey-scale image of the generated observation process. Pretending that only the observations are available, the goal is to assess the filtering distribution $p_{x^t|y^{1:t}}(x^t|y^{1:t})$ for each time step $t = 1, \dots, T$.

As described in Section 5, the assumed model $f_{x^t|\theta^t}(x^t|\theta^t)$ is a first-order Markov chain, and the parameter θ^t represents its initial and transition probabilities. Moreover, θ^t is a vector of the Dirichlet distributed random variables

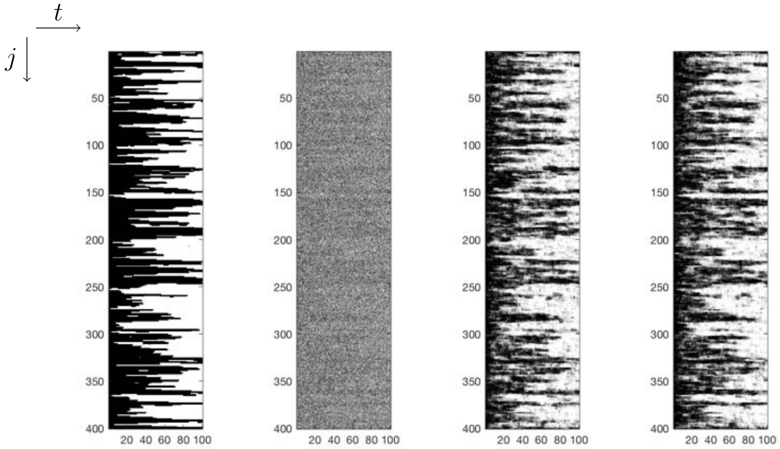
$$\theta_1^t = (\theta_1^t(0), \theta_1^t(1)),$$

$$\theta_j^{t,0} = (\theta_j^{t,0}(0), \theta_j^{t,0}(1)),$$

and

$$\theta_j^{t,1} = (\theta_j^{t,1}(0), \theta_j^{t,1}(1)),$$

for $j = 2, \dots, n$. The corresponding hyper-parameters $\alpha_1^t(0)$, $\alpha_1^t(1)$, $\alpha_j^{t,0}(0)$,



(a) (b) (c) (d)

Figure 10: First-order Markov chain simulation example: (a) the latent state process, (b) the observations, (c) estimates of marginal filtering probabilities obtained with the proposed Bayesian updating approach, and (d) estimates of marginal filtering probabilities obtained with the standard updating approach. In all figures, the colour black represents the value zero and the colour white represent the value one.

$\alpha_j^{t,0}(1)$, $\alpha_j^{t,1}(0)$, $\alpha_j^{t,1}(1)$ are all set equal to 2 at every time step t. For the assumed likelihood $f_{y^t|t}(y^t|x^t)$ we use the same distribution as the one used to simulate the data; that is, each distribution $f_{y_j^t|x_j^t}(y_j^t|x_j^t)$ is a Gaussian with mean x_j^t and variance $\sigma^2 = 2^2$. In the Gibbs simulation of $\theta^{t,(i)}|x^{t,-(i)}, y^t$, 100 iterations are used. Finally, as in Loe and Tjelmeland (2021), we use the ensemble size $M = 20$.

7.2 Simulation results

To evaluate the performance of the proposed approach, we compare our results with corresponding results obtained using the method of Loe and Tjelmeland (2021). For simplicity, we refer in the following to the method proposed in the present report as the Bayesian approach, and the method proposed in Loe and Tjelmeland (2021) as the non-Bayesian or the standard approach.

Figures 10(c) and (d) show grey-scale images of estimated values $\hat{p}(x_j^t = 1|y^{1:t})$ of the marginal filtering probabilities $p_{x_j^t|y^{1:t}}(x_j^t = 1|y^{1:t})$, $j = 1, \ldots, n$, $t = 1, \ldots, T$ obtained with the Bayesian and the non-Bayesian approach, respectively, where the estimate $\hat{p}(x_j^t = 1|y^{1:t})$ is the empirical mean of the $\tilde{x}_j^{t,(i)}$-samples,

$$\hat{p}(x_j^t = 1|y^{1:t}) = \frac{1}{M}\sum_{i=1}^{M} \tilde{x}_j^{t,(i)}. \tag{50}$$

From a visual inspection, the output from the two approaches look very similar. To investigate this further, we perform five independent runs of each method and estimate the marginal filtering probabilities in each run. For each of the two methods, we thereby obtain five samples, $\hat{p}^{(r)}(x_j^t = 1|y^{1:t})$, $r = 1, \dots, 5$, of $\hat{p}(x_j^t = 1|y^{1:t})$ in Eq. (50). Figure 11 shows plots of the empirical means of these five samples for locations $j = 1$ to 100 at the (arbitrarily chosen) time step $t = 50$, along with the corresponding minimum and maximum values of the five samples. Equivalent output from other time steps t and for other locations j follow the same trend and are therefore, for simplicity, not included. As seen in Figure 11, the results from the two methods look very much the same. This may suggest that the Bayesian approach offers no considerable advantage over the standard, non-Bayesian approach, at least not when it comes to estimating marginal filtering probabilities.

Methodologically, the main difference between the Bayesian and the non-Bayesian approach is that θ^t is treated as random in the Bayesian approach. More specifically, the Bayesian approach simulates a parameter value $\theta^{t,(i)}$ for each ensemble member $x^{t,(i)}$, while the standard approach instead computes an estimate, $\hat{\theta}^t$, and this same estimate $\hat{\theta}^t$ is used to update all the forecast samples. Therefore, since the Bayesian approach incorporates randomness in θ^t, one would expect the spread, or the variability, in the samples from the Bayesian approach to be greater than the variability in the samples from the standard approach, which is also what we observed in the simulation example with the linear-Gaussian model presented in the previous section. However, it appears that this is not the case for the binary simulation experiment studied here. For continuous variables, variability is easy to measure and visualise, but for categorical variables, other techniques are necessary. To study the variability of the results in the categorical context of this example, we consider the *coefficient of unalikeability* (CU) of Kader and Perry (2007). Given a set of independent random samples taking values in a categorical sample space, the CU provides a measure for how unalike the samples are. In the present simulation example, we are interested in computing the CU of the filtering ensemble $\{\tilde{x}^{t,(1)}, \dots, \tilde{x}^{t,(M)}\}$ at each of the time steps $t = 1, \dots, T$.

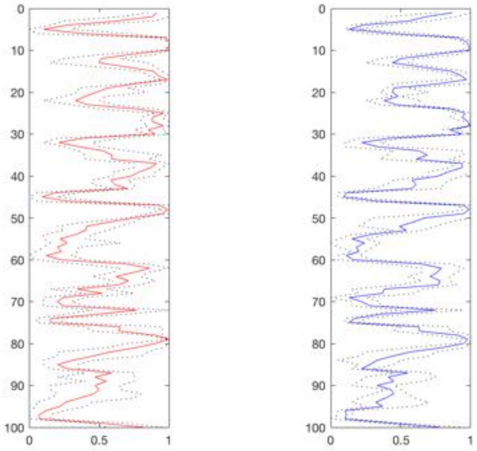


Figure 11: First-order Markov chain simulation example: The left plot shows the empirical means (solid red line) of five estimated values $\hat{p}(x_j^t = 1|y^{1:t})$ for the marginal filtering probability $p_{x_j^t|y^{1:t}}(x_j^t = 1|y^{1:t})$, $j = 1, \dots, 100$ obtained from five independent runs of the Bayesian approach, along with the corresponding minimum and maximum values (dotted black lines) of the five estimates. The right plot shows corresponding output from the standard, non-Bayesian approach.

Hereafter, we denote the CU of $\{\tilde{x}^{t,(1)}, \dots, \tilde{x}^{t,(M)}\}$ by u^t. Since $\tilde{x}^{t,(i)}$ is a vector of $n = 400$ binary variables, there are 2^{400} possible configurations for $\tilde{x}^{t,(i)}$. Each configuration can be interpreted as a (unique) category. Hence, each realisation $\tilde{x}^{t,(i)}$ of the posterior ensemble corresponds to one of the 2^{400} possible categories. However, we only have $M = 20$ ensemble members, which is not enough to give an informative value for u^t when the number of categories is so high. Therefore, we consider first each four-tuple $x_{j:j+3}^t = (x_j^t, x_{j+1}^t, x_{j+2}^t, x_{j+3}^t)$, $j = 1, \dots, n-3$, of x^t separately. The number of possible configurations for each such four-tuple is $2^4 = 16$, and from the posterior samples $\tilde{x}_{j:j+3}^{t,(1)}, \dots, \tilde{x}_{j:j+3}^{t,(M)}$ we can compute a coefficient of unalikeability u_j^t. After having computed u_j^t for each four-tuple $x_{j:j+3}^t$ of x^t, we compute the mean, $\bar{u}^t$, of all of them. This $\bar{u}^t$ then serves as an approximation for the actual CU, u^t, of $\{\tilde{x}^{t,(1)}, \dots, \tilde{x}^{t,(M)}\}$. Figure 12 shows a plot of the values of $\bar{u}^t$, $t = 1, \dots, T$, obtained with the Bayesian approach (red line) and the standard approach (blue line). As one can see, the values of $\bar{u}^t$ from the Bayesian approach very much coincide with the values from the standard approach, which indicates a similar variability in the samples.

After several additional tests, both with different data $\{y^t\}_{t=1}^T$, different values

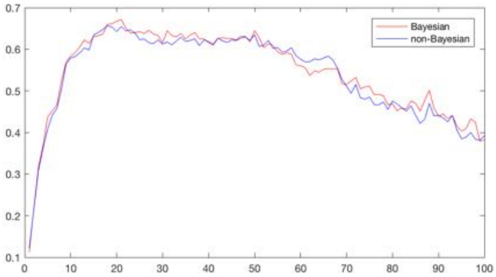


Figure 12: First-order Markov chain simulation example: Plots of the approximated coefficients of unalikeability, $\bar{u}^t$, computed at each time step $t = 1, \ldots, 100$, for the Bayesian approach (red) and the standard approach (blue).

for the observation noise σ, and different values for the ensemble size M, it seems that the variability in the results from the two approaches, and the results from the two approaches in general, are very much alike. One possible reason for this, is the optimality criterion for $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^t, y^t)$, i.e. the criterion of maximising the expected number of unchanged components of $x^{t,(i)}$. Basically, the optimality criterion states that we want to make minimal changes to the forecast samples, and this results in that the distributions $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \theta^{t,(i)}, y^t)$, $i = 1, \ldots, M$, from the Bayesian approach and the distribution $q(\tilde{x}^{t,(i)}|x^{t,(i)}, \hat{\theta}^t, y^t)$ from the standard approach are all drawn towards each other. Consequently, the generated posterior samples from the two approaches will be similar to each other. Another possible reason for the lack of differing variability is the binary nature of the problem. More specifically, since both approaches capture the mean of x_j^t quite well, they must also capture the variance, as there is a one-to-one relationship between the mean and variance for a binary random variable.

8 Closing remarks

In this report, a general framework for the updating of a prior ensemble to a posterior ensemble is presented. Being able to update a prior ensemble to a posterior ensemble is a crucial step in ensemble-based solutions to the filtering, or data assimilation, problem. The proposed method is based on an assumed Bayesian model and a proposed optimality criterion.

The general framework is investigated in two situations, one where the elements of the state vector are continuous variables and one where the elements are binary variables. In the continuous case, an assumed Gaussian distribution is adopted for the state vector and a linear-Gaussian model for the observation. This results in a class of updating methods where a fully Bayesian version of the EnKF is a special case, and we prove that a particular square root filter is optimal with respect to the optimality criterion of making minimal changes to each ensemble member. In the binary case, the state and observation vectors are instead assumed to constitute a binary HMM. The updating procedure is then essentially the same as the one for binary vectors proposed in Loe and Tjelmeland (2021), except now the transition probabilities of the assumed Markov chain model are treated as random.

When studying the results of the simulation example with the linear-Gaussian model, the most striking result is that the proposed approach is considerably better than the traditional EnKF and the hierarchical EnKF of Myrseth and Omre (2010) in representing the uncertainty. According to our analyses, the main reason for this behaviour is that we do not condition on the ensemble member which is to be updated when we simulate a corresponding vector of parameters. That we do not observe the same dramatic effect in the example with the HMM may be because in that model the same parameters control both the mean and the variance. As the non-Bayesian ensemble filtering method seems to capture the mean quite well, it must also give a good representation of the variance.

Computational efficiency is not a main focus in the present report. The dynamic programming procedure developed for the assumed HMM requires computing time proportional to the number of elements in the state vector and is thereby computationally efficient. The updating procedure of the assumed linear-Gaussian model requires inversion of $n \times n$ matrices, where n is the dimension of the state vector, so this procedure is only computationally feasible for sufficiently small values of n. In typical applications of the EnKF, the state vector is very large and computational efficiency is therefore essential. In the EnKF, the prior covariance matrix is estimated by the empirical covariance matrix of the prior ensemble. The rank of the (estimated) covariance matrix is thereby limited by the number of ensemble members, which is typically much smaller than the dimension of the state vector. The low rank of the covariance matrix makes it possible to rephrase the EnKF updating equation so that efficient computation is possible. In the proposed approach for the assumed linear-Gaussian model, the generated covariance matrices are by construction of full rank. It should, however, be possible to get computational efficiency by restricting the inverse covariance matrices, i.e. precision matrices, to be sparse. To achieve this, a prior tailored to produce sparse precision matrices must be constructed and the class of updating distributions must be restricted to ensure that all necessary computations for the updating can be performed on sparse matrices. The details of this is a direction of future research.

In the present report, we have studied in detail two applications of the proposed framework. In the future, it is of interest to explore also other assumed models and other optimality criteria. It would in particular be interesting to consider a situation where the state vector represents a two-dimensional lattice of categorical variables. A possible assumed prior model is then a Markov mesh model (Abend et al., 1965). It would also be interesting to apply the proposed framework in a mixed discrete and continuous situation, i.e. a model where the state vector consists of both discrete and continuous variables.

References

Abend, K., Harley, T., & Kanal, L. (1965). Classification of binary random patterns. *IEEE Transactions on Information Theory*, *11*, 538–544.

Anderson, J. L., & Anderson, S. L. (1999). A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, *127*, 2741–2758.

Burgers, G., van Leeuwen, P. J., & Evensen, G. (1998). Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, *126*, 1719–1724.

Doucet, A., de Freitas, N., & Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. Springer-Verlag, New York.

Evensen, G. (1994). Sequential data assimilation with a non-linear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Geophysical Research*, *99*, 10143–10162.

Evensen, G. (2003). The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, *53*, 343–367.

Frei, M., & Künsch, H. (2012). Sequential state and observation noise covariance estimation using combined ensemble Kalman and particle filters. *Monthly Weather Review*, *140*, 1476–1495.

Frei, M., & Künsch, H. (2013). Bridging the ensemble Kalman and particle filters. *Biometrika*, *100*, 781–800.

Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE-Proceedings-F*, *140*, 107–113.

Hamill, T. M., & Whitaker, J. S. (2001). Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Monthly Weather Review*, *129*, 2776–2790.

Houtekammer, P. L., & Mitchell, H. L. (2001). A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, *129*, 123–137.

Kader, G. D., & Perry, M. (2007). Variability for categorical variables [DOI: 10.1080/10691898.2007.11889465]. *Journal of Statistics Education*, *15*.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, *82*, 35–45.

Katzfuss, M., Stroud, J. R., & Wikle, C. K. (2016). Understanding the ensemble Kalman filter. *The American Statistician*, *70*, 350–357.

Katzfuss, M., Stroud, J. R., & Wikle, C. K. (2020). Ensemble Kalman methods for high-dimensional hierarchical dynamic space-time models. *Journal of the American Statistical Association*, *115*, 866–885.

Künsch, H. R. (2000). State space and hidden Markov models. In O. E. Barndorff-Nielsen, D. R. Cox, & C. Klüppelberg (Eds.), *Complex stochastic systems.* Chapman and Hall/CRC, Chap. 3, p. 109-174.

Loe, M. K., & Tjelmeland, H. (2021). Ensemble updating of binary state vectors by maximising the expected number of unchanged components [DOI: 10.1111/sjos.12483]. *Scandinavian Journal of Statistics,* To Appear.

Myrseth, I., & Omre, H. (2010). Hierarchical ensemble Kalman filter. *SPE Journal*, *15*, 569–580.

Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., Kalnay, E., Patil, D. J., & Yorke, J. A. (2004). A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, *56*, 415–428.

Stroud, J. R., Katzfuss, M., & Wikle, C. K. (2018). A Bayesian adaptive ensemble Kalman filter for sequential state and parameter estimation. *Monthly Weather Review*, *146*, 373–386.

Tippett, M. K., Anderson, J. L., Bishop, C. H., & Hamill, T. M. (2003). Ensemble square root filters. *Monthly Weather Review*, *131*, 1485–1490.

Tsyrulnikov, M., & Rakitko, A. (2017). A hierarchical Bayes ensemble Kalman filter. *Physica D: Nonlinear Phenomena*, *338*, 1–16.

A Proof of Example 3

Here, we prove the result of Example 3, i.e. that when B^t and S^t are as specified in Eqs. (28) and (29), the linear update in Eq. (27) corresponds to the stochastic EnKF update in Eq. (9).

We start by inserting the expression for B^t in Eq. (28) into Eq. (27). This gives

$$\tilde{x}^{t,(i)} = x^{t,(i)} + K^t(y^t - H^t x^{t,(i)}) + \tilde{\epsilon}^{t,(i)}. \tag{51}$$

Comparing Eq. (51) with the stochastic EnKF update in Eq. (9) we see that it remains to show that the distribution of $\tilde{\epsilon}^{t,(i)}$ in Eq. (51) is identical to the distribution of $K^t \epsilon^{t,(i)}$ in Eq. (9). As both $\tilde{\epsilon}^{t,(i)}$ and $\epsilon^{t,(i)}$ are Gaussian with zero mean, the distributions of $\tilde{\epsilon}^{t,(i)}$ and $K^t \epsilon^{t,(i)}$ are equal if

$$\mathrm{Cov}\big[\tilde{\epsilon}^{t,(i)}\big] = \mathrm{Cov}\big[K^t \epsilon^{t,(i)}\big] \,.$$

Since we have $\mathrm{Cov}[\tilde{\epsilon}^{t,(i)}] = S^t$, with S^t given by Eq. (29), and $\mathrm{Cov}[K^t \epsilon^{t,(i)}] = K^t R^t (K^t)^\top$, this means that we need to show that

$$(I_n - K^t H^t) Q^t (K^t (H^t)^\top) = K^t R^t (K^t)^\top,$$

or rather

$$(I_n - K^t H^t) Q^t (H^t)^\top = K^t R^t. \tag{52}$$

In order to prove Eq. (52) we first prove that

$$\big((Q^t)^{-1} + (H^t)^\top (R^t)^{-1} H^t\big)^{-1} (Q^t)^{-1} = I_n - K^t H^t \tag{53}$$

and

$$\big((Q^t)^{-1} + (H^t)^\top (R^t)^{-1} H^t\big)^{-1} (H^t)^\top (R^t)^{-1} = K^t. \tag{54}$$

To prove Eqs. (53) and (54) we make use of the following two formulations of the Woodbury matrix identity,

$$\big((Q^t)^{-1} + (H^t)^\top (R^t)^{-1} H^t\big)^{-1} = Q^t + Q^t (H^t)^\top \big(R^t + H^t Q^t (H^t)^\top\big)^{-1} H^t Q^t, \tag{55}$$

$$\begin{aligned}&\left(R^t + H^tQ^t(H^t)^\top\right)^{-1}\\&\qquad\qquad = (R^t)^{-1} - (R^t)^{-1}H^t\left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)^{-1}(H^t)^\top(R^t)^{-1}. \quad (56)\end{aligned}$$

To prove Eq. (53), we start by inserting Eq. (55) on the left hand side in Eq. (53) and use that the Kalman gain is given as $K^t = Q^t(H^t)^\top\left(H^tQ^t(H^t)^\top + R^t\right)^{-1}$,

$$\begin{aligned}&\left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)^{-1}(Q^t)^{-1}\\&= \Big(Q^t + Q^t(H^t)^\top\left(R^t + H^tQ^t(H^t)^\top\right)^{-1}H^tQ^t\Big)(Q^t)^{-1}\\&= I_n - K^tH^t.\end{aligned}$$

Thereby, we see that the left-hand-side and the right-hand-side in Eq. (53) are equal, and Eq. (53) is thereby proved. To prove Eq. (54) we start by considering $\left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)K^t$, insert that $K^t = Q^t(H^t)^\top\left(H^tQ^t(H^t)^\top + R^t\right)^{-1}$, and use the Woodbury identity in Eq. (56). Specifically,

$$\begin{aligned}&\left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)K^t\\&= \left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)Q^t(H^t)^\top\left(H^tQ^t(H^t)^\top + R^t\right)^{-1}\\&= \left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)Q^t(H^t)^\top\\&\qquad\cdot\Big((R^t)^{-1} - (R^t)^{-1}H^t\left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)^{-1}(R^t)^{-1}\Big)\\&= \left((H^t)^\top + (H^t)^\top(R^t)^{-1}H^tQ^t(H^t)^\top\right)\\&\qquad\cdot\left((R^t)^{-1} - (R^t)^{-1}H^t((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t)^{-1}(R^t)^{-1}\right)\\&= (H^t)^\top(R^t)^{-1} + (H^t)^\top(R^t)^{-1}H^tQ^t(H^t)^\top(R^t)^{-1}\\&\qquad - (H^t)^\top(R^t)^{-1}H^t\left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)^{-1}(H^t)^\top(R^t)^{-1}\\&\qquad - (H^t)^\top(R^t)^{-1}H^tQ^t(H^t)^\top(R^t)^{-1}H^t\left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)^{-1}(H^t)^\top(R^t)^{-1}\\&= (H^t)^\top(R^t)^{-1} - (H^t)^\top(R^t)^{-1}H^tQ^t\\&\quad\cdot\Big[-I_n + \left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)\left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)^{-1}\Big](H^t)^\top(R^t)^{-1}\\&= (H^t)^\top(R^t)^{-1}.\end{aligned}$$

Hence, we have shown that

$$\left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)K^t = (H^t)^\top(R^t)^{-1}.$$

Multiplying by $\left((Q^t)^{-1} + (H^t)^\top(R^t)^{-1}H^t\right)^{-1}$ on both sides, we get Eq. (54). Now, to prove Eq. (52) we insert Eq. (53) on the left hand side of Eq. (52) and insert

Eq. (54) on the right hand side of Eq. (52). Specifically, the left hand side of Eq. (52) then reads

$$\begin{aligned}(I_n - K^t H^t)Q^t(H^t)^\top &= \left((Q^t)^{-1} + (H^t)^\top (R^t)^{-1} H^t\right)^{-1} (Q^t)^{-1} Q^t (H^t)^\top \\ &= \left((Q^t)^{-1} + (H^t)^\top (R^t)^{-1} H^t\right)^{-1} (H^t)^\top, \qquad (57)\end{aligned}$$

while the right hand side reads

$$\begin{aligned}&\left((Q^t)^{-1} + (H^t)^\top (R^t)^{-1} H^t\right)^{-1} (H^t)^\top (R^t)^{-1} R^t \\ &\qquad = \left((Q^t)^{-1} + (H^t)^\top (R^t)^{-1} H^t\right)^{-1} (H^t)^\top. \qquad (58)\end{aligned}$$

We see that the expressions in Eqs. (57) and (58) are equal, and the proof is complete.

B Proof of Theorem 1

For any real matrices M and N of equal dimension, let $\langle M, N\rangle$ denote the Frobenius inner product,

$$\langle M, N\rangle = \mathrm{tr}(MN^\top)$$

The Cauchy-Schwarz inequality, $|\langle M, N\rangle|^2 \leq \langle M, M\rangle\langle N, N\rangle$, then gives

$$\mathrm{tr}(MN^\top)^2 \leq \mathrm{tr}(MM^\top)\mathrm{tr}(NN^\top)$$

with equality if and only if there exists a constant $c \in \mathbb{R}$ such that $M = cN$.

Using the singular value decomposition of Z, i.e. $Z = PGF^\top$, we can write

$$\mathrm{tr}(\tilde{B}Z) = \mathrm{tr}(\tilde{B}PGF^\top) = \mathrm{tr}(\tilde{B}PG^{\frac{1}{2}}(FG^{\frac{1}{2}})^\top).$$

The Cauchy-Schwarz inequality for $\mathrm{tr}\left(\tilde{B}PG^{\frac{1}{2}}(FG^{\frac{1}{2}})^\top\right)$ with $M = \tilde{B}PG^{\frac{1}{2}}$ and $N = FG^{\frac{1}{2}}$ then gives

$$\mathrm{tr}\left(\tilde{B}Z\right)^2 \quad \leq \quad \mathrm{tr}\left(\tilde{B}PG^{\frac{1}{2}}(\tilde{B}PG^{\frac{1}{2}})^\top\right)\mathrm{tr}\left(FG^{\frac{1}{2}}(FG^{\frac{1}{2}})^\top\right) \qquad (59)$$

with equality if and only if there exists a number $c \in \mathbb{R}$ such that

$$\tilde{B}PG^{\frac{1}{2}} = cFG^{\frac{1}{2}} \iff \tilde{B} = cFP^\top.$$

Using basic trace properties and that $\tilde{B}^\top \tilde{B} = I_n - \tilde{S}$ and $F^\top F = P^\top P = I_n$, the right hand side in (59) can be rewritten as

$$\begin{aligned}
&\operatorname{tr}\left(\tilde{B}PG^{\frac{1}{2}}(\tilde{B}PG^{\frac{1}{2}})^\top\right)\operatorname{tr}\left(FG^{\frac{1}{2}}(FG^{\frac{1}{2}})^\top\right)\\
&\quad= \operatorname{tr}\left(\tilde{B}PGP^\top\tilde{B}^\top\right)\operatorname{tr}\left(FGF^\top\right)\\
&\quad= \operatorname{tr}\left(PGP^\top\tilde{B}^\top\tilde{B}\right)\operatorname{tr}\left(GF^\top F\right)\\
&\quad= \operatorname{tr}\left(PGP^\top(I-\tilde{S})\right)\operatorname{tr}\left(G\right)\\
&\quad= \left(\operatorname{tr}\left(PGP^\top\right) - \operatorname{tr}\left(PGP^\top\tilde{S}\right)\right)\operatorname{tr}(G)\\
&\quad= \left(\operatorname{tr}(G) - \operatorname{tr}(PGP^\top\tilde{S})\right)\operatorname{tr}(G).
\end{aligned}$$

When $\tilde{S} = 0$, we see that the Cauchy-Schwarz inequality yields

$$\operatorname{tr}(\tilde{B}Z)^2 \leq \operatorname{tr}(G)^2$$

with equality if and only if there exists $c \in \mathbb{R}$ such that $\tilde{B} = cFP^\top$. The condition that $\tilde{S} = I_n - \tilde{B}^\top\tilde{B} = 0$ gives restrictions on the allowed values for c. Specifically,

$$\begin{aligned}
I_n - \tilde{B}^\top\tilde{B} &= I_n - (cFP^\top)^\top(cFP^\top) = I_n - c^2PF^\top FP^\top\\
&= (1-c^2)I_n = 0 \iff c = \pm 1.
\end{aligned}$$

Hence, when $\tilde{S} = 0$, the maximum value of $\operatorname{tr}(\tilde{B}Z)^2$ is $\operatorname{tr}(G)^2$ and this occurs only for $\tilde{B} = \pm FP^\top$. The maximum value of $\operatorname{tr}(\tilde{B}Z)$ is thereby $\operatorname{tr}(G)$ which occurs when $c = 1$, i.e. for $\tilde{B} = FP^\top$.

When $\tilde{S} \neq 0$, we need to study the sign of $\operatorname{tr}\left(PGP^\top\tilde{S}\right)$. Since G is a diagonal matrix we get

$$\operatorname{tr}\left(PGP^\top\tilde{S}\right) = \operatorname{tr}\left(GP^\top\tilde{S}P\right) = \sum_{i=1}^{n} G_{ii}(P^\top\tilde{S}P)_{ii}.$$

We have assumed Z to have full rank, so all singular values of Z are strictly positive, i.e. $G_{ii} > 0$ for each i. Let $\tilde{S}$ have singular value decomposition $\tilde{S} =$

$WJW^\top$. We then get

$$\begin{aligned}(P^\top \tilde{S} P)_{ii} &= \left(P^\top W J W^\top P\right)_{ii} = \left((W^\top P)^\top J W^\top P\right)_{ii} \\ &= \sum_{k=1}^{n} J_{kk} \left(W^\top P\right)_{ki}^2 .\end{aligned}$$

Since we have assumed $\tilde{S} \neq 0$ at least one of the singular values of $\tilde{S}$ must be strictly positive, i.e. we have at least one $J_{kk} > 0$. Without loss of generality we assume in the following that $J_{11} > 0$. Since both P and W are orthogonal matrices $W^\top P$ is also orthogonal. Thereby there exists at least one index i such that $(W^\top P)_{1i} > 0$. For this value of i we then have

$$(P^\top \tilde{S} P)_{ii} \geq J_{11} (W^\top P)_{1i}^2 > 0.$$

Thereby, since $P^\top \tilde{S} P$ is positive semidefinite,

$$\operatorname{tr}\left(P G P^\top \tilde{S}\right) \geq G_{ii} (P^\top \tilde{S} P)_{ii} > 0.$$

Thus,

$$|\operatorname{tr}(\tilde{B} Z)| \leq \sqrt{\left(\operatorname{tr}(G) - \operatorname{tr}(P G P^\top \tilde{S})\right) \operatorname{tr}(G)} < \operatorname{tr}(G).$$

We thereby see that the maximum value of $\operatorname{tr}(\tilde{B} Z)$ when $\tilde{S} \neq 0$ is *smaller* than its maximum value when $\tilde{S} = 0$. The maximum value of $\operatorname{tr}(\tilde{B} Z)$ must therefore occur when $\tilde{S} = 0$ and $\tilde{B} = F P^\top$, and the proof is complete.

Paper IV

Ensemble updating of categorical state vectors

Margrethe Kvale Loe and Håkon Tjelmeland

Submitted

Ensemble updating of categorical state vectors

Margrethe Kvale Loe

Department of Mathematical Sciences, Norwegian University of Science and Technology

Håkon Tjelmeland

Department of Mathematical Sciences, Norwegian University of Science and Technology

Abstract

An ensemble updating method for categorical state vectors is presented. The method is based on a Bayesian and generalised view of the ensemble Kalman filter (EnKF). In the EnKF, Gaussian approximations to the forecast and filtering distributions are introduced, and the forecast ensemble is updated with a linear shift. Given that the Gaussian approximation to the forecast distribution is correct, the EnKF linear update corresponds to conditional simulation from a Gaussian distribution with mean and covariance such that the posterior samples marginally are distributed according to the Gaussian approximation to the filtering distribution. In the proposed approach for categorical vectors, the Gaussian approximations are replaced with other parametric models, appropriate for the categorical context, and instead of a linear update, we characterise, for each forecast ensemble member, a class of decomposable graphical models (DGMs) for simulating a corresponding posterior ensemble member. To make the update robust against the assumptions of the assumed forecast and filtering distributions, an optimality criterion is formulated. The proposed framework is Bayesian in the sense that the parameters of the assumed forecast distribution are treated as random. We study in detail the proposed framework when a (possibly higher-order) Markov chain is adopted for the forecast model. The optimal DGM can then be constructed by solving a linear program. A simulation example where each variable of the state vector can take three different values is presented.

Keywords: Bayesian statistics; Data assimilation; Ensemble updating; Markov chains

1 Introduction

State-space models are widely used to analyse dynamic data in a broad range of scientific disciplines, e.g. in finance, reservoir modelling, weather forecasting, and signal processing. A general state-space model consists of an unobserved process $\{x^t\}_{t=1}^T$ and a corresponding observed process $\{y^t\}_{t=1}^T$ where y^t is a partial observation of x^t. The unobserved x^t-process, usually called the state process, is assumed to be a first-order Markov process, and the observations $y^1, \dots, y^T$ of the observed process are assumed to be conditionally independent given $\{x^t\}_{t=1}^T$ with y^t only depending on x^t. The main objective of state-space modelling is some type of inference about the state process given the observations. There are many inference procedures associated with state-space models, among which one of the most common is filtering. Filtering, which is the problem addressed in the present article, refers to the task of computing, for each time step $t = 1, \dots, T$, the distribution of the state x^t given all observations $y^{1:t} = (y^1, \dots, y^t)$ available at time t. In some fields, filtering is known as sequential data assimilation. Other common terms are history matching and online inference. However, in the present article, we use the term filtering throughout.

Because of the particular dependency structure of the general state-space model, the series of filtering distributions can be computed recursively according to a recursion which alternates between a forecast step and an update step. Generally, however, apart from a few simple special cases, the exact solution to the filtering recursions is intractable due to complex and/or high-dimensional integrals. Approximate strategies are therefore required, and simulation-based methods, or ensemble methods, represent the most popular approach. An ensemble-based solution may, similarly to the original filtering recursions, alternate between a forecast step and an update step. Instead, however, of computing the forecast and filtering distributions explicitly, the distributions are represented empirically with an ensemble of realisations. The main challenge in this context is the update step where, at time step t, an ensemble of (approximate) realisations from the so-called forecast distribution $p_{x^t|y^{1:t-1}}(x^t|y^{1:t-1})$ needs to be conditioned on the new observation y^t so that an updated ensemble of (approximate) realisations from the filtering distribution $p_{x^t|y^{1:t}}(x^t|y^{1:t})$ is obtained. Since there is no straightforward way to approach this task, ensemble methods require approximations in the update step. This *ensemble updating problem* is the core focus of the present paper. In particular, we address the problem of updating an ensemble of *categorical* state vectors and we present in detail an approximate ensemble updating method for this situation.

Among the ensemble filtering methods that have currently been proposed in the literature there are two main categories; particle filters (Gordon et al., 1993; Doucet et al., 2001) and ensemble Kalman filters (EnKFs) (Burgers et al., 1998; Evensen, 2003; Tippett et al., 2003). Particle filters are based on importance sampling while EnKFs rely on a linear-Gaussian assumption about the underlying state-space model. Particle filters have the advantage of being asymptotically exact in the sense that as the ensemble size goes to infinity, the filters converge to the exact filtering solution. In practical applications, however, computational resources often restrict the ensemble size to be quite small, and particle filters are known to collapse unless the ensemble size is very large compared to the state dimension (Snyder et al., 2008). For the EnKF, the solution is always biased unless the underlying state-space model really is linear-Gaussian. Despite this fact, the EnKF often performs remarkably well also in non-linear, non-Gaussian situations and, unlike the particle filter, also scales well to problems with very high-dimensional states. The filter is, however, inappropriate in situations with categorical vectors, as considered in the present paper.

Loe and Tjelmeland (2021a), in a follow-up study to Loe and Tjelmeland (2021b), present an alternative solution to the ensemble updating problem based on a generalised view of the EnKF. Specifically, they describe a general updating framework where the idea is to first introduce assumed models for the intractable forecast and filtering distributions and thereafter to update the prior samples by simulating samples from a distribution which, under the assumption that the assumed forecast distribution is correct, preserves the corresponding assumed filtering distribution. To make the update robust against the assumptions of the assumed forecast and filtering models, the distribution from which the posterior samples are simulated is also required to be optimal with respect to a chosen optimality criterion. More specifically, the updating distribution is required to minimise the expected value of a certain distance, or norm, between a prior (forecast) and posterior (filtering) ensemble member. The framework is also Bayesian in the sense that the parameters of the forecast distribution are treated as random variables. Uncertainty about these parameters are thereby incorporated into the updating. Two particular applications of the proposed framework are investigated, one continuous and one categorical. In the continuous case, the assumed forecast and filtering models are chosen as Gaussian distributions and the optimality criterion is to minimise the expected Mahalanobis distance between a prior and posterior ensemble member. The framework then leads to a Bayesian version of square root EnKF (Bishop et al., 2001; Whitaker and Hamill, 2002; Tippett et al., 2003). In the categorical case, the assumed forecast and filtering distributions are instead chosen as first-order Markov chains and the optimality criterion is to minimise the expected number of variables of a prior state vector that change their values. An optimal transition matrix for simulating a posterior ensemble member from a corresponding prior ensemble member is constructed using a combination of dynamic and linear programming.

There are three important limitations about the updating procedure for categorical state vectors proposed in Loe and Tjelmeland (2021a). Firstly, the procedure is difficult to implement except in the binary case where there are only two possible values for each variable of the state vector. Consequently, the authors only demonstrate the method in binary numerical experiments. Secondly, the approximation to the forecast distribution is restricted to be a first-order Markov chain. This means that models with higher-order interactions, for example a higher-order Markov chain, cannot be considered. Thirdly, the procedure is not applicable in two- or three-dimensional problems since it requires that the state vector has a one-dimensional spatial arrangement. In the present article, we address the first and second of these three issues. Specifically, we present a modified and improved version of the updating procedure applicable also for $K > 2$ classes and which allows the use of a higher-order Markov chain as the approximate forecast distribution. In the procedure described in Loe and Tjelmeland (2021a), a model with respect to a directed acyclic graph (DAG) is put forward to update each forecast realisation. The chosen structure of the DAG allows the corresponding optimal updating distribution to be computed recursively using a dynamic programming algorithm where a piecewise-linear programming problem is solved in each recursive step. In the present article, the starting point is, instead of a DAG model, an undirected graphical model. By choosing the underlying graph as decomposable (Cowell et al., 1999), we get a model with many convenient computational properties, and the optimal updating distribution can be computed by solving a linear program derived from a series of local computations on the undirected graphical model.

The remains of this paper take the following outline. In Section 2, we review state-space models and the associated filtering problem in more detail, and we also present some basic graph theory required to understand the proposed approach. In Section 3, we present a slightly modified version of the general ensemble updating framework in Loe and Tjelmeland (2021a), restricting the focus to categorical state vectors. In Section 4, we describe in detail how the general framework can be applied when a Markov chain model, possibly of higher order, is adopted for the assumed forecast distribution. Thereafter, we present in Section 5 a simulation example where each element of the state vector can take $K = 3$ values. Finally, we finish off with a few closing remarks in Section 6.

2 Preliminaries

This section describes the filtering problem in more detail and also reviews some graph-theoretic concepts related to the proposed approach. The section also introduces notations that we use throughout the paper.

2.1 The filtering problem

A general state-space model consists of an unobserved process $\{x^t\}_{t=1}^T, x^t \in \Omega_x$, called the state process, and a corresponding observed process $\{y^t\}_{t=1}^T, y^t \in \Omega_y$, called the observation process, where y^t is a partial observation of x^t at time t. The unobserved state process $\{x^t\}_{t=1}^T$ is modelled as a first-order Markov chain with initial distribution $p_{x^1}(x^1)$ and transition probabilities $p_{x^t|x^{t-1}}(x^t|x^{t-1})$, $t = 2, \ldots, T$. Throughout this paper, we use the notations $x^{s:t} = (x^s, \ldots, x^t)$ and $y^{s:t} = (y^s, \ldots, y^t)$, $s \leq t$, to denote the vector of all states and the vector of all observations, respectively, from time s to time t. The joint distribution of $x^{1:T}$ follows from the first-order Markov assumptions as

$$p_{x^{1:T}}(x^{1:T}) = p_{x^1}(x^1) \prod_{t=2}^{T} p_{x^t|x^{t-1}}(x^t|x^{t-1}).$$

For the observation process, it is assumed that $y^1, \ldots, y^T$ are conditionally independent given $x^{1:T}$, with y^t only depending on x^t. The conditional distribution of $y^{1:T}$ given $x^{1:T}$ thereby follows as

$$p_{y^{1:T}|x^{1:T}}(y^{1:T}|x^{1:T}) = \prod_{t=1}^{T} p_{y^t|x^t}(y^t|x^t).$$

It is possible to adjust the state-space model formulated above so that observations are only recorded at a subset of the time steps from 1 to T. However, for simplicity, we assume in this work that an observation is recorded at every time step $t = 1, \ldots, T$.

The objective of the filtering problem is, for each time step $t = 1, \ldots, T$, to compute the so-called filtering distribution, $p_{x^t|y^{1:t}}(x^t|y^{1:t})$, i.e. the distribution of the latent state x^t given all the observations $y^{1:t}$ available at time t. Because of the particular structure of the state-space model, the series of filtering distributions can be computed recursively according to a recursion which alternates between a forecast step,

$$p_{x^t|y^{1:t-1}}(x^t|y^{1:t-1}) = \int_{\Omega_x} p_{x^t|x^{t-1}}(x^t|x^{t-1}) p_{x^{t-1}|y^{1:t-1}}(x^{t-1}|y^{1:t-1}) \mathrm{d}x^{t-1}, \tag{1}$$

and an update step,

$$p_{x^t|y^{1:t}}(x^t|y^{1:t}) = \frac{p_{x^t|y^{1:t-1}}(x^t|y^{1:t-1}) p_{y^t|x^t}(y^t|x^t)}{p_{y^t|y^{1:t-1}}(y^t|y^{1:t-1})}, \tag{2}$$

where

$$p_{y^t|y^{1:t-1}}(y^t|y^{1:t-1}) = \int_{\Omega_x} p_{x^t|y^{1:t-1}}(x^t|y^{1:t-1}) p_{y^t|x^t}(y^t|x^t) \mathrm{d}x^t. \tag{3}$$

The distribution $p_{x^t|y^{1:t-1}}(x^t|y^{1:t-1})$ computed in the forecast step is called the forecast distribution of x^t. In the update step, this distribution is conditioned on the new observation y^t in order to compute the filtering distribution of x^t, $p_{x^t|y^{1:t}}(x^t|y^{1:t})$. The update step is essentially a standard Bayesian inference problem with the forecast distribution becoming the prior and the filtering distribution the posterior.

There are two important special cases where the filtering recursions can be computed exactly. The first is the linear-Gaussian model where the initial distribution $p_{x^1}(x^1)$ is Gaussian and where $p_{x^t|x^{t-1}}(x^t|x^{t-1})$ and $p_{y^t|x^t}(y^t|x^t)$ are Gaussian with mean vectors being linear functions of x^{t-1} and x^t, respectively. The forecast and filtering distributions are then also Gaussian and Eqs. (1) and (2) lead to the famous Kalman filter (Kalman, 1960). The second situation where the filtering recursions are tractable is the finite state-space hidden Markov model for which the state-space Ω_x consists of a finite number of values. The integrals in Eqs. (1) and (3) then reduce to finite sums. If, however, the number of states in Ω_x is large, for example if x^t is a high-dimensional vector of categorical variables, the summations become too computer-demanding to cope with, and the filtering recursions are left computationally intractable.

Generally, the integrals in Eqs. (1) and (3) make the recursive solution to the filtering problem intractable, and approximate solutions therefore become necessary. The most popular approach is the class of ensemble-based methods, where a set of samples, an ensemble, is used to empirically represent the sequence of filtering distributions. A great advantage of the ensemble context is that it simplifies the forecast step. Specifically, if an ensemble $\{z^{t,(1)}, \ldots, z^{t,(M)}\}$ of independent realisations from the filtering distribution $p_{x^t|y^{1:t}}(x^t|y^{1:t})$ is available, a forecast ensemble $\{x^{t+1,(1)}, \ldots, x^{t+1,(M)}\}$ with independent realisations from the forecast distribution $p_{x^{t+1}|y^{1:t}}(x^{t+1}|y^{1:t})$ can be obtained by simulating

$$x^{t+1,(i)}|z^{t,(i)} \sim p_{x^{t+1}|x^t}(\cdot|z^{t,(i)})$$

independently for $i = 1, \ldots, M$. The consecutive updating of the ensemble, however, remains challenging. There is simply no straightforward way to condition the forecast ensemble $\{x^{t+1,(1)}, \ldots, x^{t+1,(M)}\}$ on the new observation y^{t+1} so that a new filtering ensemble $\{z^{t+1,(1)}, \ldots, z^{t+1,(M)}\}$ of independent realisations from the filtering distribution $p_{x^{t+1}|y^{1:t+1}}(x^{t+1}|y^{1:t+1})$ is obtained. In the present article, we propose an approximate way to do this when the elements of the state vector are categorical variables.

2.2 Decomposable graphical models (DGMs)

This section introduces decomposable graphical models (DGMs), a certain type of undirected graphical models, or Markov random fields (Kindermann and Snell, 1980; Cressie, 1993; Cowell et al., 1999). For simplicity, the focus is restricted to discrete DGMs. In the following, we start with a brief review of some basic theory on undirected graphs in Sections 2.2.1 and 2.2.2. Thereafter, discrete DGMs are introduced in Section 2.2.3, while Sections 2.2.4 and 2.2.5 consider simulation from discrete DGMs. A more thorough introduction to graph theory and graphical models can be found in, e.g., Cowell et al. (1999).

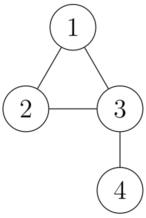


Figure 1: An undirected graph with four vertices

2.2.1 Undirected graphs

An undirected graph G is an ordered pair $G = (V, E)$ where V is a set of vertices, or nodes, and $E \subset \{V \times V\}$ is a set of edges. The elements of the edge set E are pairs of distinct nodes, $\{i, j\}$, $i, j \in V$, $i \neq j$. If $\{i, j\} \in E$ then node i and node j are said to be *neighbours*, or adjacent. Figure 1 illustrates a simple undirected graph with four vertices where, as per convention, vertices are represented by labelled circles and edges by lines between the circles. For this graph we have $V = \{1, 2, 3, 4\}$ and $E = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{3, 4\}\}$.

If there is an edge between every pair of nodes in a graph G, the graph is said to be *complete*. A *subgraph* of G is a graph $G_A = (A, E_A)$ where $A \subseteq V$ and $E_A \subseteq E \ \cap \{A \times A\}$. If a subgraph $G_A = (A, E_A)$ of G is complete, its set of nodes A is called a *clique*. A clique is called a *maximal clique* in G if it is not a subset of another clique. Throughout this article, we denote the set of maximal cliques by C. For the graph pictured in Figure 1, the empty set $\emptyset$ and $\{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{3, 4\}$, and $\{1, 2, 3\}$ are cliques, while $\{1, 2, 3\}$ and $\{3, 4\}$ are maximal cliques.

A *path* of length n from node i to node j is a sequence $(\alpha_0, \ldots, \alpha_n)$ of distinct nodes where $\alpha_0 = i$ and $\alpha_n = j$ and $\{\alpha_{k-1}, \alpha_k\} \in E$, $k = 1, \ldots, n$. Note that this means that if there is a path from node i to node j in an undirected graph, there is also a path from node j to node i. For the graph pictured in Figure 1, there are two paths from node 1 to node 4: $(1, 2, 3, 4)$ and $(1, 3, 4)$. Two nodes i and j are said to be *connected* if there is a path from node i to node j, and an undirected graph is said to be connected if every pair of vertices are connected. A *tree* is a connected undirected graph with the additional property that the path between every pair of vertices is unique. The graph in Figure 1 is thus not a tree since there are different paths between some of the vertices.

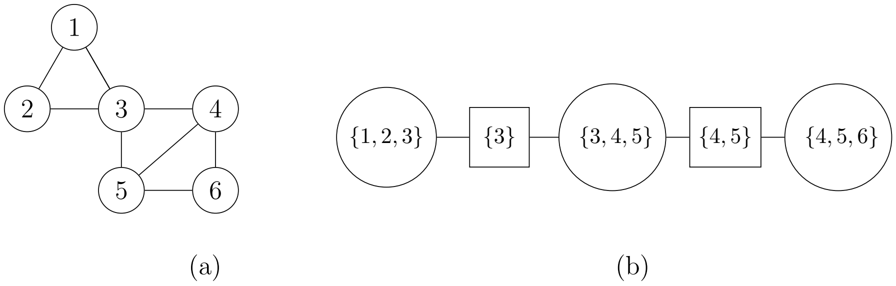


Figure 2: (a) A decomposable graph, and (b) a corresponding junction tree representation

2.2.2 Decomposable graphs and junction trees

An undirected graph $G = (V, E)$ is decomposable if the set of maximal cliques $C = \{c_1, \ldots, c_{|C|}\}$, where $|C|$ denotes the number of maximal cliques, can be ordered as $(c_1, \ldots, c_{|C|})$ so that for each $i = 1, \ldots, |C| - 1$ there is a $j > i$ such that

$$s_i = c_i \cap (c_{i+1} \cup \cdots \cup c_{|C|}) \subseteq c_j. \tag{4}$$

The property in Eq. (4) is called *the running intersection property*, and the sets $s_1, \ldots,\ s_{|C|-1}$ are called the *separators* of the graph. The set of all separators, $S = \{s_1, \ldots, s_{|C|-1}\}$, and the set of maximal cliques, C, are uniquely determined by the structure of the graph G; however, the ordering $(c_1, \ldots, c_{|C|})$ is generally not unique. Figure 2(a) shows a simple decomposable graph with six vertices. The maximal cliques of this graph are $\{1, 2, 3\}$, $\{3, 4, 5\}$ and $\{4, 5, 6\}$, and the separators are $\{3\}$ and $\{4, 5\}$. If we order the maximal cliques as $(c_1, c_2, c_3) = (\{1, 2, 3\}, \{3, 4, 5\}, \{4, 5, 6\})$, we find the corresponding ordering (s_1, s_2) of the separators as $s_1 = \{3\} = \{1, 2, 3\} \cap (\{3, 4, 5\} \cup \{4, 5, 6\}) \subseteq c_2$ and $s_2 = \{4, 5\} = \{3, 4, 5\} \cap \{4, 5, 6\} \subseteq c_3$.

From a decomposable graph, a corresponding *junction tree* can be derived. A junction tree J for a decomposable graph G is a tree with $C = \{c_1, \ldots, c_{|C|}\}$ as its node set and the additional property that for every pair $c_i, c_j \in C$ every node on the unique path between c_i and c_j in J contains the intersection $c_i \cap c_j$. In a visual representation of a junction tree, it is common to include the separators as squared labels on the edges. This is illustrated in Figure 2(b) which shows one of the possible junction tree representations for the decomposable graph in Figure 2(a).

A junction tree is a nice way to organise a decomposable graph, and many computations are easier to perform on the junction tree. Depending on the structure of the graph, however, it can be a complicated task to construct a corresponding junction tree. There exist several algorithms for this purpose, see Cowell et al. (1999). In all the examples we encounter in the present article, the decomposable graphs have a structure which makes it particularly simple to construct junction trees, and therefore we do not focus on the problem of constructing junction trees in this paper.

2.2.3 Discrete decomposable graphical models

A discrete decomposable graphical model (DGM) is a probabilistic model consisting of a decomposable graph $G = (V, E)$, a random vector $x = (x_i, i \in V)$ of categorical variables $x_i \in \{0, 1, \ldots, K-1\}$, and a probability distribution $p_x(x)$. Alternatively, a discrete DGM can be defined as a discrete Markov random field whose underlying graph is decomposable. In the following, the notation x_A is used to denote the variables of x associated with the subset $A \subseteq V$, and $\Omega_{x_A} \subseteq \Omega_x$ denotes the sample space of x_A. Taking $0/0 = 0$, the distribution $p_x(x)$ of a discrete DGM can be expressed as

$$p_x(x) = \frac{\prod_{c \in C} p_{x_c}(x_c)}{\prod_{s \in S} p_{x_s}(x_s)}, \tag{5}$$

where C is the set of maximal cliques in G and S is the set of separators (Cowell et al., 1999). DGMs support several efficient algorithms and are fundamental for the work of this article. In particular, it should be noted that, if $(c_1, \ldots, c_{|C|})$ is an ordering of the maximal cliques fulfilling the running intersection property in Eq. (4) and $(s_1, \ldots, s_{|C|-1})$ is the corresponding ordering of the separators, we have

$$p_{x_{s_i}}(x_{s_i}) = \sum_{x_{c_i \setminus s_i}} p_{x_{c_i}}(x_{c_i}) = \sum_{x_{c_{i+1} \setminus s_i}} p_{x_{c_{i+1}}}(x_{c_{i+1}}) \quad \text{for all } x_{s_i} \in \Omega_{x_{s_i}}. \tag{6}$$

2.2.4 Simulation from discrete DGMs

Consider a discrete DGM $p_x(x)$ with respect to a graph $G = (V, E)$. To simulate a realisation from $p_x(x)$, a recursive procedure can be adopted, which goes as follows. First, $p_x(x)$ is decomposed into $p_{x_i|x_{V \setminus \{i\}}}(x_i|x_{V \setminus \{i\}})$ and $p_{x_{V \setminus \{i\}}}(x_{V \setminus \{i\}})$ for some $i \in V$. Thereafter, $p_{x_{V \setminus \{i\}}}(x_{V \setminus \{i\}})$ is decomposed into $p_{x_j|x_{V \setminus \{i,j\}}}(x_j|x_{V \setminus \{i,j\}})$ and $p_{x_{V \setminus \{i,j\}}}(x_{V \setminus \{i,j\}})$ for some $j \in V \setminus \{i\}$. Then, $p_{x_{V \setminus \{i,j\}}}(x_{V \setminus \{i,j\}})$ is decomposed into $p_{x_k|x_{V \setminus \{i,j,k\}}}(x_k|x_{V \setminus \{i,j,k\}})$ and $p_{x_{V \setminus \{i,j,k\}}}(x_{V \setminus \{i,j,k\}})$ for some $k \in V \setminus \{i, j\}$. Continuing in this manner, we ultimately end up with only one variable x_l and corresponding marginal distribution $p_{x_l}(x_l)$. A realisation $x \sim p_x(\cdot)$ can then be generated by recursively simulating from the series of conditional distributions, in the reverse order as they were computed. Without loss of generality, suppose that the vertex set is $V = \{1, \ldots, n\}$ and that the nodes have been numbered so that nodes are removed in the order from n to 1. This means that we make us of the following factorisation of $p_x(x)$:

$$p_x(x) = p_{x_1}(x_1) \prod_{i=2}^{n} p_{x_i|x_{1:i-1}}(x_i|x_{1:i-1}). \tag{7}$$

Having computed all the factors in Eq. (7), simulation from $p_x(x)$ follows easily by first simulating $x_1 \sim p_{x_1}(\cdot)$, thereafter $x_2|x_1 \sim p_{x_2|x_1}(\cdot|x_1)$, and so on. The recursive procedure described above, as well as the factorisation in Eq. (7), is general and holds for any distribution $p_x(x)$, not necessarily a discrete DGM. However, for many models, it is not convenient to factorise $p_x(x)$ in this manner, since it can be a complicated task to compute all the factors. If the model is a DGM, however, and a corresponding junction tree J is available, computations become particularly easy and efficient, as we discuss in the following.

First, note that the distribution in Eq. (5) can be expressed as

$$p_x(x) \propto \exp\left\{ \sum_{c \in C} V_c(x_c) \right\}, \tag{8}$$

where $V_c(x_c)$ in this context is called a potential function for clique c. With the junction tree J given, it is convenient to start the decomposition of $p_x(x)$ in a leaf of J. Denote the clique to which the chosen leaf corresponds by c^*. Since c^* is a leaf of J, there is at least one node $i \in V$ which is only present in c^*. Suppose, without loss of generality, that the nodes have been numbered so that this is the case for node n, i.e. that node n is only contained in clique c^*. We can then easily decompose $p_x(x)$ into $p_{x_n|x_{1:n-1}}(x_n|x_{1:n-1})$ and $p_{x_{1:n-1}}(x_{1:n-1})$ as follows. Since node n is only contained in clique c^*, the variable x_n only enters the right-hand-side expression in Eq. (8) through the potential function $V_{c^*}(x_{c^*})$. This means that $p_{x_n|x_{1:n-1}}(x_n|x_{1:n-1})$ can be computed as

$$p_{x_n|x_{1:n-1}}(x_n|x_{1:n-1}) = \frac{\exp\{V_{c^*}(x_{c^*})\}}{\sum_{x_n} \exp\{V_{c^*}(x_{c^*})\}}. \tag{9}$$

The other part, $p_{x_{1:n-1}}(x_{1:n-1})$, can be computed, up to a constant of proportionality, by summing out x_n from Eq. (8),

$$p(x_{1:n-1}) \propto \sum_{x_n} \left(\exp\left\{ \sum_{c \in C \setminus c^*} V_c(x_c) \right\} \exp\left\{ V_{c^*}(x_{c^*}) \right\} \right).$$

Using that node n is only contained in clique c^*, we can rewrite this expression as

$$p_{x_{1:n-1}}(x_{1:n-1}) \propto \exp\left\{ \sum_{c \in C \setminus c^*} V_c(x_c) \right\} \sum_{x_n} \exp\left\{ V_{c^*}(x_{c^*}) \right\}. \qquad (10)$$

That is, we only need to sum over x_n in $\exp\{V_{c^*}(x_{c^*})\}$. Now, if we define a new potential function for the clique $c^* \setminus \{n\}$,

$$V_{c^* \setminus \{n\}}(x_{c^* \setminus \{n\}}) = \log\left(\sum_{x_n} \exp\left\{ V_{c^*}(x_{c^*}) \right\} \right),$$

we can rewrite Eq. (10) in the more convenient form

$$p_{x_{1:n-1}}(x_{1:n-1}) \propto \exp\left\{ \sum_{c \in C \setminus c^*} V_c(x_c) \right\} \exp\left\{ V_{c^* \setminus \{n\}}(x_{c^* \setminus \{n\}}) \right\}. \qquad (11)$$

It is not necessary to compute the normalising constant in Eq. (11) in order for the remaining computations to proceed.

Next, we must split $p_{x_{1:n-1}}(x_{1:n-1})$ into $p_{x_{n-1}|x_{1:n-2}}(x_{n-1}|x_{1:n-2})$ and $p_{x_{1:n-2}}(x_{1:n-2})$. For this, consider first the junction tree $J_{V \setminus \{n\}}$ we obtain after removing node n from c^* in J. Removing node n from c^* can affect the structure of $J_{V \setminus \{n\}}$ in two different ways: either $J_{V \setminus \{n\}}$ has the same number of nodes as J, or it has one node less. To understand why, consider the clique $c^* \setminus \{n\}$ that we obtain after removing node n from c^*. Moreover, let $\tilde{c}$ denote the neighbour of c^* in J and let $G_{V \setminus \{n\}}$ denote the graph obtained by removing node n from G. For the clique $c^* \setminus \{n\}$, there are now two possibilities: either it is a subset of $\tilde{c}$, i.e. $c^* \setminus \{n\} \subseteq \tilde{c}$, or it is *not* a subset of $\tilde{c}$, i.e. $c^* \setminus \{n\} \nsubseteq \tilde{c}$. If $c^* \setminus \{n\} \nsubseteq \tilde{c}$, then $c^* \setminus \{n\}$ is a maximal clique in the graph $G_{V \setminus \{n\}}$, and $J_{V \setminus \{n\}}$ is essentially the same tree as J except that c^* is replaced with $c^* \setminus \{n\}$. The clique $c^* \setminus \{n\}$ then represents a leaf in $J_{V \setminus \{n\}}$, and we can decompose $p_{x_{1:n-1}}(x_{1:n-1})$ into $p_{x_{n-1}|x_{1:n-2}}(x_{n-1}|x_{1:n-2})$

and $p_{x_{1:n-2}}(x_{1:n-2})$ in the same manner as we decomposed $p_x(x)$ above. If, on the other hand, $c^* \setminus \{n\} \subseteq \tilde{c}$, we must merge $c^* \setminus \{n\}$ and $\tilde{c}$ before we can proceed. Specifically, this entails that we need to define a new clique potential for $\tilde{c}$, namely as the sum of the potential function for $c^* \setminus \{n\}$ and the current potential function for $\tilde{c}$,

$$\widetilde{V}_{\tilde{c}}(x_{\tilde{c}}) = V_{\tilde{c}}(x_{\tilde{c}}) + V_{c^*\setminus\{n\}}(x_{c^*\setminus\{n\}}).$$

We can then rewrite Eq. (11) as

$$p_{x_{1:n-1}}(x_{1:n-1}) \propto \exp\left\{ \sum_{c \in C\setminus\{c^*,\tilde{c}\}} V_c(x_c) \right\} \exp\left\{ \widetilde{V}_{\tilde{c}}(x_{\tilde{c}}) \right\}. \tag{12}$$

After merging the cliques, we can decompose $p_{x_{1:n-1}}(x_{1:n-1})$ in Eq. (12) into $p_{x_{1:n-2}}(x_{1:n-2})$ and $p_{x_{n-1}|x_{1:n-2}}(x_{n-1}|x_{1:n-2})$ in the same manner as we decomposed $p_x(x)$ into $p_{x_{1:n-1}}(x_{1:n-1})$ and $p_{x_n|x_{1:n-1}}(x_n|x_{1:n-1})$ above. Notice, however, that it is possible that $\tilde{c}$ is not a leaf in $J_{V\setminus\{n\}}$. If so, we must move to a clique which does represent a leaf, and decompose $p_{x_{1:n-1}}(x_{1:n-1})$ by removing a node and corresponding variable from this clique.

Ultimately, we end up computing $p_{x_1}(x_1)$. A realisation from $p_x(x)$ can then be obtained by first simulating $x_1 \sim p_{x_1}(\cdot)$, thereafter $x_2|x_1 \sim p_{x_2|x_1}(\cdot|x_1)$, then $x_3|x_1, x_2 \sim p_{x_3|x_1,x_2}(\cdot|x_1, x_2)$, and so on.

2.2.5 Conditional simulation from discrete DGMs

Suppose again that $p_x(x)$ is a discrete DGM with respect to a graph $G = (V, E)$, and let J be a junction tree for G. In the previous section, we described how to simulate from $p_x(x)$. Now, we address the closely related problem of how to simulate from the conditional distribution $p_{x_A|x_{V\setminus A}}(x_A|x_{V\setminus A})$, $A \subset V$. First, note that

$$p_{x_A|x_{V\setminus A}}(x_A|x_{V\setminus A}) \propto p_{x_A,x_{V\setminus A}}(x_A, x_{V\setminus A}) = p_x(x). \tag{13}$$

By inserting values for $x_{V\setminus A}$ in Eq. (13), we obtain an expression for $p_{x_A|x_{V\setminus A}}(x_A|x_{V\setminus A})$ up to a constant of proportionality. Thus, since $p_{x_A|x_{V\setminus A}}(x_A|x_{V\setminus A})$ is also a discrete DGM, we can simulate from $p_{x_A|x_{V\setminus A}}(x_A|x_{V\setminus A})$ using the recursive procedure described in Section 2.2.4, as this procedure only requires that $p_{x_A|x_{V\setminus A}}(x_A|x_{V\setminus A})$ is known up to a constant of proportionality. Before starting the computations, however, a new graph G_A and corresponding junction tree J_A must be constructed for $p_{x_A|x_{V\setminus A}}(x_A|x_{V\setminus A})$, and the clique potentials for

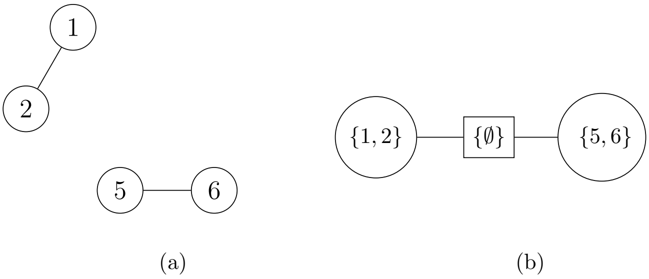


Figure 3: (a) The subgraph G_A with $A = \{1, 2, 5, 6\}$ for the graph in Figure 2, and (b) the corresponding junction tree J_A

the maximal cliques of G_A must be computed. The graph G_A is simply obtained by removing the nodes $V \setminus A$ from V and all edges $\{i, j\}$ from E where $i \in V \setminus A$ and/or $j \in V \setminus A$.

As an illustrative example, consider a DGM with respect to the graph in Figure 2(a). Suppose values for x_3 and x_4 are given and that we want to simulate from the conditional distribution $p(x_1, x_2, x_5, x_6|x_3, x_4)$,

$$\begin{aligned} &p_{x_1,x_2,x_5,x_6|x_3,x_4}(x_1, x_2, x_5, x_6|x_3, x_4) \\ &\qquad \propto \exp\Big\{ V_{\{1,2,3\}}(x_1, x_2, x_3) + V_{\{3,4,5\}}(x_3, x_4, x_5) + V_{\{4,5,6\}}(x_4, x_5, x_6) \Big\}. \end{aligned}$$

For this toy example, we have $A = \{1, 2, 5, 6\}$ and $V \setminus A = \{3, 4\}$. The graph G_A is shown in Figure 3(a) and the junction tree J_A is shown in Figure 3(b). The graph G_A only has two maximal cliques, $\{1, 2\}$ and $\{5, 6\}$, and the separator is simply the empty set $\emptyset$. The potential functions corresponding to the maximal cliques $\{1, 2\}$ and $\{5, 6\}$ become, respectively,

$$\widetilde{V}_{\{1,2\}}(x_1, x_2) = V_{\{1,2,3\}}(x_1, x_2, x_3)$$

and

$$\widetilde{V}_{\{5,6\}}(x_5, x_6) = V_{\{3,4,5\}}(x_3, x_4, x_5) + V_{\{4,5,6\}}(x_4, x_5, x_6),$$

where now x_3 and x_4 are constant values. With G_A, J_A and these potential functions given, we can simulate from

$$p_{x_1,x_2,x_5,x_6|x_3,x_4}(x_1, x_2, x_5, x_6|x_3, x_4) \propto \exp\Big\{ \widetilde{V}_{\{1,2\}}(x_1, x_2) + \widetilde{V}_{\{5,6\}}(x_5, x_6) \Big\}$$

using the procedure described in Section 2.2.4.

3 General updating framework for categorical vectors

In this section, we describe a general ensemble updating framework for categorical state vectors. The framework is a slightly modified version of the framework presented in Loe and Tjelmeland (2021a) and involves the following steps. First, for each sample $x^{t,(i)}$ of the prior ensemble, a Bayesian model is adopted for the updating of $x^{t,(i)}$ to $z^{t,(i)}$. Then, a class of 'updating distributions', specifically a class of DGMs, for simulating the posterior sample $z^{t,(i)}$ conditionally on $x^{t,(i)}$, is characterised. Finally, an optimality criterion is formulated so that a corresponding optimal updating distribution can be computed. The optimality criterion is chosen to make the updating robust against the assumptions of the assumed Bayesian model.

3.1 Assumed Bayesian model

To update the forecast ensemble, we propose to update each forecast sample $x^{t,(i)}$ separately and to adopt a Bayesian model for this update. Figure 4 shows a graphical illustration of the assumed Bayesian model that we adopt for the updating of $x^{t,(i)}$. The assumed Bayesian model includes an unknown parameter vector $\theta^t \in \Omega_\theta$ for which a prior model $f_{\theta^t}(\theta^t)$ is adopted. Moreover, the latent state vector x^t and the prior samples $x^{t,(1)}, \ldots, x^{t,(M)}$ are all assumed to be conditionally independent and identically distributed given θ^t, i.e.

$$f_{x^t, x^{t,(1)}, \ldots, x^{t,(M)}|\theta^t}(x^t, x^{t,(1)}, \ldots, x^{t,(M)}|\theta^t) = f_{x^t|\theta^t}(x^t|\theta^t) \prod_{i=1}^{M} f_{x^t|\theta^t}(x^{t,(i)}|\theta^t),$$

where $f_{x^t|\theta^t}(x^t|\theta^t)$ is an assumed prior model for x^t given θ^t. The observation y^t is assumed to be conditionally independent of θ^t and $x^{t,(1)}, \ldots, x^{t,(M)}$ given x^t, and distributed according to an assumed likelihood model $f_{y^t|x^t}(y^t|x^t)$. Given $x^{t,(i)}$, θ^t and y^t, the posterior realisation $z^{t,(i)}$ is conditionally independent of $x^{t,(1)}, \ldots, x^{t,(i-1)}, x^{t,(i+1)}, \ldots, x^{t,(M)}$ and x^t. For simplicity, we denote in the following the set of prior samples except the sample $x^{t,(i)}$ by $x^{t,-(i)}$,

$$x^{t,-(i)} = \{x^{t,(1)}, \ldots, x^{t,(i-1)}, x^{t,(i+1)}, \ldots, x^{t,(M)}\}.$$

Conceptually, the assumed models $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{y^t|x^t}(y^t|x^t)$ can be any

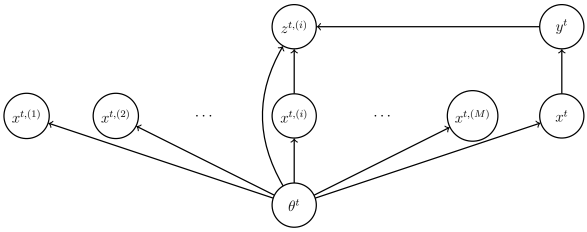


Figure 4: Graphical illustration of the assumed Bayesian model for the updating of $x^{t,(i)}$

parametric distributions. In order for the framework to be useful in practice, however, they must be chosen so that the corresponding posterior model

$$f_{x^t|\theta^t,y^t}(x^t|\theta^t,y^t) \propto f_{x^t|\theta^t}(x^t|\theta^t)f_{y^t|x^t}(y^t|x^t) \tag{14}$$

is tractable. Moreover, $f_{\theta^t}(\theta^t)$ should be chosen as conjugate for $f_{x^t|\theta^t}(x^t|\theta^t)$.

3.2 Class of updating distributions

Based on the Bayesian model introduced above, we characterise in this section a class of updating distributions for generating $z^{t,(i)}$ from $x^{t,(i)}$. First, we derive in Section 3.2.1 a class of updating distributions which are exact in the sense that, under the assumption that the forecast samples are distributed according to the assumed prior model $f_{x^t|\theta^t}(x^t|\theta^t)$, the posterior sample $z^{t,(i)}$ is distributed according to the corresponding assumed posterior model in Eq. (14). Thereafter, we introduce in Section 3.2.2 a class of approximate updating distributions which can be easier to deal with in practice.

3.2.1 Derivation of a class of updating distributions

A natural minimal restriction for the updating of $x^{t,(i)}$ to $z^{t,(i)}$ is to require that the procedure is consistent with the assumed model. One can then say that the updating is correct under the assumed model. In addition to this first restriction, one would also like the updating to be robust against the assumptions made in the assumed model.

A naïve updating procedure that is consistent with the assumed model is simply to set $z^{t,(i)}$ equal to a sample from $f_{x_t|x^{t,(1)},\ldots,x^{t,(M)},y^t}(\cdot|x^{t,(1)},\ldots,x^{t,(M)},y^t)$. This procedure may, however, be very sensitive to the assumptions of the assumed model. To get a more robust updating procedure, a better alternative is to generate $z^{t,(i)}$ as a modified version of $x^{t,(i)}$, as indicated by the graph in Figure 4. In such a setup, the role of $x^{t,(i)}$ is as a source of randomness in the generation of $z^{t,(i)}$. One should therefore remove $x^{t,(i)}$ from the conditioning set in the naïve updating procedure and instead require that $z^{t,(i)}$ is a sample from $f_{x^t|x^{t,-(i)},y^t}(\cdot|x^{t,-(i)},y^t)$ under the assumed model. Thus, the updating of $x^{t,(i)}$ to $z^{t,(i)}$ should be such that

$$f_{z^{t,(i)}|x^{t,-(i)},y_t}(x^t|x^{t,-(i)},y^t) = f_{x^t|x^{t,-(i)},y^t}(x^t|x^{t,-(i)},y^t) \tag{15}$$

for all x^t, $x^{t,-(i)}$ and y^t. Exactly how the generation of $z^{t,(i)}$ from $x^{t,(i)}$ should be performed to get an updating procedure that is robust against an error in the assumed model is discussed in Section 3.3. In the following, we first focus on the implications of the restriction in Eq. (15) under the assumed Bayesian model illustrated in Figure 4.

Having introduced the parameter θ^t, the criterion in Eq. (15) can be rewritten as

$$\int_{\Omega_{\theta^t}} f_{\theta^t,z^{t,(i)}|x^{t,-(i)},y^t}(\theta^t,x^t|x^{t,-(i)},y^t)\mathrm{d}\theta^t = \int_{\Omega_{\theta^t}} f_{\theta,x^t|x^{t,-(i)},y^t}(\theta^t,x^t|x^{t,-(i)},y^t)\mathrm{d}\theta^t.$$

This can further be rewritten as

$$\int_{\Omega_{\theta^t}} f_{\theta^t|x^{t,-(i)},y^t}(\theta^t|x^{t,-(i)},y^t) f_{z^{t,(i)}|\theta^t,y^t}(x^t|\theta^t,y^t)\mathrm{d}\theta^t = \int_{\Omega_{\theta^t}} f_{\theta^t|x^{t,-(i)},y^t}(\theta^t|x^{t,-(i)},y^t) f_{x^t|\theta^t,y^t}(x^t|\theta^t,y^t)\mathrm{d}\theta^t. \tag{16}$$

A sufficient condition for Eq. (16) to hold is

$$f_{z^{t,(i)}|\theta^t,y^t}(x^t|\theta^t,y^t) = f_{x^t|\theta^t,y^t}(x^t|\theta^t,y^t) \tag{17}$$

for all x^t,θ^t and y^t. Thereby, we understand that $x^{t,(i)}$ can be updated by first simulating

$$\theta^{t,(i)}|x^{t,-(i)},y^t \sim f_{\theta^t|x^{t,-(i)},y^t}(\cdot|x^{t,-(i)},y^t)$$

and thereafter simulate

$$z^{t,(i)}|x^{t,(i)},\theta^{t,(i)},y^t \sim f_{z^{t,(i)}|x^{t,(i)},\theta^{t,(i)},y^t}(\cdot|x^{t,(i)},\theta^{t,(i)},y^t),$$

where $f_{z^{t,(i)}|x^{t,(i)},\theta^{t,(i)},y^t}(z^{t,(i)}|x^{t,(i)},\theta^{t,(i)},y^t)$ is a distribution which fulfils Eq. (17).

Generally, a class of updating distributions $f_{z^{t,(i)}|x^{t,(i)},\theta^{t,(i)},y^t}(z^{t,(i)}|x^{t,(i)},\theta^{t,(i)},y^t)$ consistent with the requirement in Eq. (17) exists. The simplest option is to use the assumed posterior model $f_{x^t|\theta^{t,(i)},y^t}(x^t|\theta^{t,(i)},y^t)$ and simulate $z^{t,(i)}$ independently of $x^{t,(i)}$. However, this means that we possibly loose valuable information from $x^{t,(i)}$ about the true forecast and filtering distributions that we may not have been able to capture with the assumed model. To preserve more of this information from $x^{t,(i)}$, it is important to simulate $z^{t,(i)}$ conditionally on $x^{t,(i)}$.

Conceptually, an updating distribution $f_{z^{t,(i)}|x^{t,(i)},\theta^{t,(i)},y^t}(z^{t,(i)}|x^{t,(i)},\theta^{t,(i)},y^t)$ can be constructed by first constructing a joint distribution $f_{x^{t,(i)},z^{t,(i)}|\theta^{t,(i)},y^t}(x^{t,(i)},z^{t,(i)}|\theta^{t,(i)},y^t)$, and thereafter condition this distribution on $x^{t,(i)}$. This joint distribution can be factorised as

$$\begin{aligned}&f_{x^{t,(i)},z^{t,(i)}|\theta^{t,(i)},y^t}(x^{t,(i)},z^{t,(i)}|\theta^{t,(i)},y^t)\\&\qquad= f_{x^t|\theta^{t,(i)}}(x^{t,(i)}|\theta^{t,(i)})f_{z^{t,(i)}|x^{t,(i)},\theta^{t,(i)},y^t}(z^{t,(i)}|x^{t,(i)},\theta^{t,(i)},y^t). \qquad (18)\end{aligned}$$

To be consistent with the requirement in Eq. (17), the joint distribution in Eq. (18) must fulfil

$$\sum_{x^{t,(i)}} f_{x^{t,(i)},z^{t,(i)}|\theta^{t,(i)},y^t}(x^{t,(i)},z^{t,(i)}|\theta^{t,(i)},y^t) = f_{x^t|\theta^{t,(i)},y^t}(z^{t,(i)}|\theta^{t,(i)},y^t), \qquad (19)$$

that is, when marginalising out $x^{t,(i)}$ we end up with the assumed posterior model. Moreover, to be consistent with the assumed Bayesian model, and so that the factorised form in Eq. (18) holds, the distribution must also fulfil

$$\sum_{z^{t,(i)}} f_{x^{t,(i)},z^{t,(i)}|\theta^{t,(i)},y^t}(x^{t,(i)},z^{t,(i)}|\theta^{t,(i)},y^t) = f_{x^t|\theta^{t,(i)}}(x^{t,(i)}|\theta^{t,(i)}), \qquad (20)$$

that is, when marginalising out $z^{t,(i)}$ we end up with the the assumed prior model. In principle, infinitely many distributions $f_{x^{t,(i)},z^{t,(i)}|\theta^{t,(i)},y^t}(x^{t,(i)},z^{t,(i)}|\theta^{t,(i)},y^t)$ consistent with the requirements in Eqs. (19) and (20) may exist. In practice, however, it is generally difficult to assess one of these distributions, except the naïve solution discussed above where $f_{z^{t,(i)}|x^{t,(i)},\theta^{t,(i)},y^t}(\cdot|x^{t,(i)},\theta^{t,(i)},y^t)$ is set equal to the assumed posterior model $f_{x^t|\theta^{t,(i)},y^t}(\cdot|\theta^{t,(i)},y^t)$. Therefore, we must resort to approximations, which we consider in more detail below.

3.2.2 A class of approximate updating distributions

In this section, we introduce approximations to the criteria in Eqs. (19) and (20) and characterise a corresponding class of DGMs which fulfil these modified requirements. We denote in the following a distribution within this class by $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$, where the semicolon does not represent a conditioning, but that the distribution is a function of $\theta^{t,(i)}$ and y^t.

Let $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$ be a DGM with respect to a graph G with vertex set $V = \{1, \ldots, 2n\}$ and maximal clique set $C = \{c_1, \ldots, c_{|C|}\}$ where $|C|$ is the number of maximal cliques. Associate the n variables of $x^{t,(i)}$ with the nodes $1, \ldots, n$ and the n variables of $z^{t,(i)}$ with the nodes $n+1, \ldots, 2n$, so that, for $j = 1, \ldots, n$, the variable $x_j^{t,(i)}$ is associated with node j and the variable $z_j^{t,(i)}$ is associated with node $j+n$. Next, let $A_1, \ldots, A_{|C|}$, $B_1, \ldots, B_{|C|}$ denote a sequence of subsets of $V_{1:n} = \{1, \ldots, n\}$ such that the nodes of V that are associated with $(x_{A_j}^{t,(i)}, z_{B_j}^{t,(i)})$ form clique c_j. Mathematically, that is

$$A_j = \{i \in c_j; i \leq n\} \tag{21}$$

and

$$B_j = \{i - n; i \in c_j, i > n\}. \tag{22}$$

Thereby, $q(x_{A_j}^{t,(i)}, z_{B_j}^{t,(i)}; \theta^{t,(i)}, y^t)$ represents the distribution of the variables $(x_{A_j}^{t,(i)}, z_{B_j}^{t,(i)})$ associated with clique c_j, $j = 1, \ldots, |C|$. For example, if $c_1 = \{1, 2, n+1\}$, then $A_1 = \{1, 2\}$ and $B_1 = \{1\}$, and $q(x_{1:2}^{t,(i)}, z_1^{t,(i)}; \theta^{t,(i)}, y^t)$ represents the distribution for the variables $(x_1^{t,(i)}, x_2^{t,(i)}, z_1^{t,(i)})$ associated with the nodes of clique c_1.

From Section 2.2 we know that since $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$ is a DGM it is fully specified by its clique probabilities and can be expressed as

$$q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t) = \frac{\prod_{j=1}^{|C|} q(x_{A_j}^{t,(i)}, z_{B_j}^{t,(i)}; \theta^{t,(i)}, y^t)}{\prod_{j=1}^{|C|-1} q(x_{A_j \cap A_{j+1}}^{t,(i)}, z_{B_j \cap B_{j+1}}^{t,(i)}; \theta^{t,(i)}, y^t)}.$$

Hence, in order to specify $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$, all we need to do is to appropriately specify each of the clique probabilities $q(x_{A_j}^{t,(i)}, z_{B_j}^{t,(i)}; \theta^{t,(i)}, y^t)$, $j = 1, \ldots, |C|$. Recall that the goal is to specify $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$ such that it approximately represents the joint distribution in Eq. (18) subject to the constraints in Eqs. (19) and (20). To construct such a $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$, we replace the requirements in Eqs. (19) and (20) by

$$\sum_{x_{A_j}^{t,(i)}} q(x_{A_j}^{t,(i)}, z_{B_j}^{t,(i)}; \theta^{t,(i)}, y^t) = f_{x_{B_j}^t|\theta^{t,(i)},y^t}(z_{B_j}^{t,(i)}|\theta^{t,(i)}, y^t), \quad j = 1, \dots, |C|, \tag{23}$$

and

$$\sum_{z_{B_j}^{t,(i)}} q(x_{A_j}^{t,(i)}, z_{B_j}^{t,(i)}; \theta^{t,(i)}, y^t) = f_{x_{A_j}^t|\theta^{t,(i)},y^t}(x_{A_j}^{t,(i)}|\theta^{t,(i)}), \quad j = 1, \dots, |C|, \tag{24}$$

respectively. That is, instead of requiring that $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$ fully preserves the assumed models $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{x^t|\theta^t}(x^t|\theta^t, y^t)$, as required in Eqs. (19) and (20), we only require that the marginal distributions $f_{x_{A_j}^t|\theta^t}(x_{A_j}^t|\theta^t)$ and $f_{x_{B_j}^t|\theta^t,y^t}(x_{B_j}^t|\theta^t, y^t)$ are preserved.

Another constraint we need to take into account when specifying $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$ is that the clique probabilities must be consistent in the sense that, if we let $(c_1, \dots, c_{|C|})$ denote an ordering of the cliques in C which fulfils the running intersection property in Eq. (4), the probabilities for two consecutive cliques c_j and c_{j+1} must return the same marginal distribution for the separator $s_j = c_j \cap c_{j+1}$. Mathematically, this can be written as

$$\begin{aligned}\sum_{x_{A_j\backslash\{A_j\cap A_{j+1}\}}^{t,(i)}} \sum_{z_{B_j\backslash\{B_j\cap B_{j+1}\}}^{t,(i)}} & q(x_{A_j}^{t,(i)}, z_{B_j}^{t,(i)}; \theta^{t,(i)}, y^t) \\ &= \sum_{x_{A_{j+1}\backslash\{A_j\cap A_{j+1}\}}^{t,(i)}} \sum_{x_{A_{j+1}\backslash\{A_j\cap A_{j+1}\}}^{t,(i)}} q(x_{A_j}^{t,(i)}, z_{B_j}^{t,(i)}; \theta^{t,(i)}, y^t). \end{aligned}\tag{25}$$

Assuming we are able to construct a DGM $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$ consistent with the requirements discussed above, we can condition this DGM on $x^{t,(i)}$ and simulate $z^{t,(i)}|x^{t,(i)}$ as described in Section 2.2.5.

3.3 Defining an optimal solution

There may be infinitely many distributions $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$ which fulfil the requirements in Eqs. (23) to (25). For us, however, it is sufficient with *one* solution and preferably an optimal solution. To preserve as much information from $x^{t,(i)}$ as possible, we propose to define the optimal solution as the solution which maximises the expected number of variables of $x^{t,(i)}$ that remain unchanged,

Algorithm 1: Summary of general updating procedure

```
1. Select the distributions $f_{\theta^t}(\theta^t)$, $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{y^t|x^t}(y^t|x^t)$ introduced in
   Section 3.1
2. for $i = 1, \dots, M$ do
       a) Simulate
              $\theta^{t,(i)}|x^{t,-(i)}, y^t \sim f_{\theta^t|x^{t,-(i)},y^t}(\cdot|x^{t,-(i)}, y^t)$
          as described in Appendix A
       b) Construct a DGM $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$ which fulfils Eqs. (23) to (25) and
          maximises Eq. (26)
       c) Simulate
              $z^{t,(i)}|x^{t,(i)} \sim q(z^{t,(i)}|x^{t,(i)}; \theta^{t,(i)}, y^t)$
          as described in Section 2.2.5
   end
```

i.e. the solution which maximises the function

$$g(x^{t,(i)}, z^{t,(i)}) = \mathrm{E}\left[\sum_{j=1}^{n} 1\left(x_j^{t,(i)} = z_j^{t,(i)}\right)\right], \tag{26}$$

where the expectation is taken over $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$. Intuitively, this seems like a reasonable optimality criterion which should make the update robust with respect to the assumed models $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t)$. By making minimal changes to $x^{t,(i)}$, the updated sample $z^{t,(i)}$ may be able to capture properties of the true filtering distribution $p_{x^t|y^{1:t}}(x^t|y^{1:t})$ that we may not have captured with the assumed posterior model $f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t)$. The key steps of our general updating procedure are summarised in Algorithm 1.

4 Updating procedure with a Markov chain assumed prior

In this section, we consider the general framework described in Section 3 when the vector $x^t = (x_1^t, \dots, x_n^t)$ is restricted to have a spatial arrangement in one-dimensional space (i.e., along a line) and a ν'th order Markov chain model is adopted for $f_{x^t|\theta^t}(x^t|\theta^t)$. For this situation, we propose to choose the maximal cliques of the DGM $q(x^{t,(i)}, z^{t,(i)}; \theta^t, y^t)$ such that the optimal solution can be computed by solving a linear optimisation problem.

4.1 Model specifications

Assuming the vector x^t has a one-dimensional spatial arrangement, we propose to choose $f_{x^t|\theta^t}(x^t|\theta^t)$ as a Markov chain of order $\nu \geq 1$,

$$f_{x^t|\theta^t}(x^t|\theta^t) = f_{x^t_{1:\nu}|\theta^t}(x^t_{1:\nu}|\theta^t) \prod_{j=\nu+1}^{n} f_{x^t_j|x^t_{j-\nu:j-1},\theta^t}(x^t_j|x^t_{j-\nu:j-1},\theta^t).$$

For the likelihood model $f_{y^t|x^t}(y^t|x^t)$, we assume that $y^t = (y^t_1, \ldots, y^t_n)$ contains n conditionally independent observations, with y^t_j depending only on x^t_j,

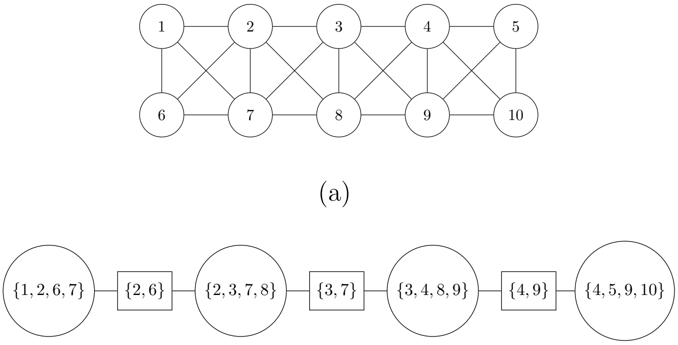
$$f_{y^t|x^t}(y^t|x^t) = \prod_{j=1}^{n} f(y^t_j|x^t_j). \tag{27}$$

This choice of prior and likelihood yields a posterior model $f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t)$ which is also a Markov chain of order ν. The initial and transition probabilities of this posterior Markov chain can be computed with a forward-backward recursive procedure (e.g., Künsch, 2000).

A natural interpretation of θ^t in this context is that it represents the initial and transition probabilities of the assumed prior Markov chain. For a Markov chain of order ν, there are $n - \nu + 1$ transition matrices to specify, each matrix consisting of K^ν rows and K columns. Denote in the following these transition matrices by $\theta^t_1, \ldots, \theta^t_{n-\nu+1}$. Furthermore, let θ_0 be a vector representing the initial probabilities of the Markov chain, and consider

$$\theta^t = (\theta^t_0, \theta^t_1, \ldots, \theta^t_{n-\nu+1}).$$

Following the recommendations of Section 3.1, we choose $f_{\theta^t}(\theta^t)$ as conjugate for $f_{x^t|\theta^t}(x^t|\theta^t)$. Here, this entails adopting a Dirichlet distribution for θ^t_0 and a Dirichlet distribution for each of the K^ν row vectors in each transition matrix θ^t_j, $j = 1, \ldots, n-\nu+1$, and to let all these Dirichlet distributed parameters be a priori independent. For simplicity, the remaining technical details of the specification of $f_{\theta^t}(\theta^t)$ are presented in Appendix A. In the same appendix, it is also described how to simulate $\theta^t|x^{t,-(i)}, y^t \sim f_{\theta^t|x^{t,-(i)},y^t}(\theta^t|x^{t,-(i)}, y^t)$.



(b)

Figure 5: (a) Underlying graph for the DGM $q(x^{t,(i)}, z^{t,(i)}; \theta^t, y^t)$ of Section 4 when $d = 2$ and $n = 5$, (b) a corresponding junction tree representation

4.2 Class of updating distributions

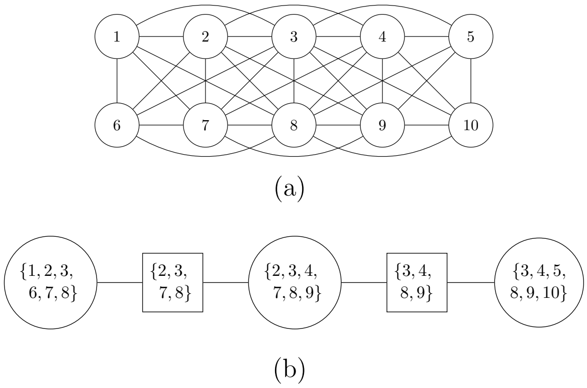
Having specified the distributions $f_{\theta^t}(\theta^t)$, $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{y^t|x^t}(y^t|x^t)$ of the assumed Bayesian model, the next task is to characterise the class of DGMs $q(x^{t,(i)}, z^{t,(i)}; \theta^t, y^t)$ introduced in Section 3.2. For this, we need to specify the cliques of the underlying decomposable graph of $q(x^{t,(i)}, z^{t,(i)}; \theta^t, y^t)$ or, equivalently, the A_j and B_j-sets in Eqs. (21) and (22). For some integer $d \geq 1$, we specify A_j and B_j as

$$A_j = B_j = \{j, j+1, \ldots, j+d-1\} \tag{28}$$

for $j = 1, \ldots, n-d+1$. Visually, the decomposable graph G can then be represented as a two-dimensional grid with two rows and n columns, or as a $2 \times n$ matrix. The first row is associated with the nodes $1, \ldots, n$ and the second row is associated with the nodes $n+1, \ldots, 2n$. Each maximal clique is formed by d consecutive columns, hence we call it a $2 \times d$ clique. The variables associated with each $2 \times d$ clique are $x^{t,(i)}_{j:j+d-1}$ and $z^{t,(i)}_{j:j+d-1}$. Figures 5(a) and 6(a) illustrate G when $d = 2$ and $d = 3$, respectively, when the state vector x^t contains $n = 5$ variables. Figures 5(b) and 6(b) show corresponding junction tree representations. The structure of G makes it fairly trivial to construct corresponding junction trees.

The criteria in Eqs. (23) to (25) can now be rewritten as

$$\sum_{x^{t,(i)}_{j:j+d-1}} q(x^{t,(i)}_{j:j+d-1}, z^{t,(i)}_{j:j+d-1}; \theta^{t,(i)}, y^t) = f_{x^t_{j:j+d-1}|\theta^{t,(i)}, y^t}(z^{t,(i)}_{j:j+d-1}|\theta^{t,(i)}, y^t), \tag{29}$$



(a)

(b)

Figure 6: (a) Underlying graph for the DGM $q(x^{t,(i)}, z^{t,(i)}; \theta^t, y^t)$ of Section 4 when $d = 3$ and $n = 5$, (b) a corresponding junction tree representation

$$\sum_{z^{t,(i)}_{j:j+d-1}} q(x^{t,(i)}_{j:j+d-1}, z^{t,(i)}_{j:j+d-1}; \theta^{t,(i)}, y^t) = f_{x^t_{j:j+d-1}|\theta^{t,(i)}}(x^{t,(i)}_{j:j+d-1}|\theta^{t,(i)}), \tag{30}$$

and

$$\sum_{x^{t,(i)}_j} \sum_{z^{t,(i)}_j} q(x^{t,(i)}_{j:j+d-1}, z^{t,(i)}_{j:j+d-1}; \theta^{t,(i)}, y^t) = \sum_{x^{t,(i)}_{j+d}} \sum_{x^{t,(i)}_{j+d}} q(x^{t,(i)}_{j+1:j+d}, z^{t,(i)}_{j+1:j+d}; \theta^{t,(i)}, y^t), \tag{31}$$

respectively.

4.3 Computing the optimal solution

When the maximal cliques of $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$ are as specified in Section 4.2, the optimal solution of $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$, i.e. the solution which maximises the expected value in Eq. (26), can be computed by solving a linear optimisation problem where the unknowns are all the clique probabilities $q(x^{t,(i)}_{A_j}, z^{t,(i)}_{A_j}; \theta^t, y^t)$, $j = 1, \ldots, n-d+1$. To see this, notice first that the objective function in Eq. (26) can be written as

$$\mathrm{E}\left[\sum_{j=1}^{n} 1\left(x^{(i)}_j = z^{(i)}_j\right)\right] = \sum_{k=0}^{K-1} \sum_{j=1}^{n} q(x^{t,(i)}_j = k, z^{t,(i)}_j = k;\ \theta^{t,(i)}, y^t).$$

Further,

$$\mathrm{E}\left[\sum_{j=1}^{n} 1\left(x_j^{(i)} = z_j^{(i)}\right)\right] = \sum_{k=0}^{K-1}\sum_{j=1}^{n-d} q(x_j^{t,(i)} = k, z_j^{t,(i)} = k; \theta^{t,(i)}, y^t) \\ + \sum_{k=0}^{K-1}\sum_{j=n-d+1}^{n} q(x_j^{t,(i)} = k, z_j^{t,(i)} = k; \theta^{t,(i)}, y^t).$$

All the terms in this expression can be computed by summing out variables from a corresponding clique distribution $q(x_{A_j}^{t,(i)}, z_{A_j}^{t,(i)}; \theta^{t,(i)}, y^t)$. More precisely, term number j in the sum from 1 to $n-d$ can be computed by summing out variables from $q(x_{A_j}^{t,(i)}, z_{A_j}^{t,(i)}; \theta^{t,(i)}, y^t)$, while each term in the sum from $n-d+1$ to n can be computed by summing out variables from $q(x_{A_{n-d+1}}^{t,(i)}, z_{A_{n-d+1}}^{t,(i)}; \theta^{t,(i)}, y^t)$. This leads to an objective function which is a linear function of $q(x_{A_j}^{t,(i)}, z_{A_j}^{t,(i)}; \theta^{t,(i)}, y^t)$, $j = 1, \ldots, n-d+1$. The objective function is to be maximised subject to the constraints in Eqs. (29) to (31), which are also linear functions of $q(x_{A_j}^{t,(i)}, z_{A_j}^{t,(i)}; \theta^{t,(i)}, y^t)$. Because $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$ is a probability distribution, we must also include the constraint that $q(x_{A_j}^{t,(i)}, z_{A_j}^{t,(i)}; \theta^{t,(i)}, y^t)$ sums to one,

$$\sum_{x_{A_j}^{t,(i)}}\sum_{z_{A_j}^{t,(i)}} q(x_{A_j}^{t,(i)}, z_{A_j}^{t,(i)}; \theta^{t,(i)}, y^t) = 1, \tag{32}$$

and that it can only take values between zero and one,

$$0 \leq q(x_{A_j}^{t,(i)}, z_{A_j}^{t,(i)}; \theta^{t,(i)}, y^t) \leq 1. \tag{33}$$

These constraints are also linear functions of $q(x_{A_j}^{t,(i)}, z_{A_j}^{t,(i)}; \theta^{t,(i)}, y^t)$. Thus, we have a linear optimisation problem, or a linear program, which can be efficiently solved with standard linear programming techniques.

5 Simulation example

In this section, the updating procedure described in Section 4 is demonstrated in a simulation example. The example involves a filtering problem where the unobserved Markov process $\{x^t\}_{t=1}^T$ consists of $T = 100$ time steps, the dimension of x^t is $n = 200$, and there are three classes for each element x_j^t of x^t: 0, 1, and 2.

5.1 Experimental setup

To specify an initial distribution $p_{x^1}(x^1)$ and a forward model $p_{x^t|x^{t-1}}(x^t|x^{t-1})$ for the latent Markov process $\{x^t\}_{t=1}^T$, we use a modified version of the binary simulation example in Loe and Tjelmeland (2021b). As for the example in that article, we let us inspire from the process when water comes through to an oil producing well in a petroleum reservoir. It should be stressed, however, that we do not claim that our model is really realistic for such a process. The t in x_j^t then represents time and j the location in the well, with $j = 1$ being at the top of the well and $j = n$ at the bottom. We let the events $x_j^t = 0$ and $x_j^t = 1$ represent the presence of porous sand stone filled with oil and water, respectively, in location j of the well at time t, while the event $x_j^t = 2$ represents non-porous shale in the same location. One should note that the spatial distribution of sand stone and shale does not change with time, whereas the fluid in a sand stone may change. Therefore, if $x_j^{t-1} = 2$, the forward model should be specified so that also $x_j^t = 2$ with probability 1, and correspondingly, if $x_j^{t-1} = 0$ or $x_j^{t-1} = 1$, the forward model should have probability zero for $x_j^t = 2$. In the start, $t = 1$, we want oil to be present in all the sand stone. Thereafter, water should gradually displace the oil and at time $t = T$ water should be the dominating fluid.

To simplify the specification of the forward model, we let x^t given x^{t-1} be a first-order Markov chain, so that

$$p_{x^t|x^{t-1}}(x^t|x^{t-1}) = p_{x_1^t|x^{t-1}}(x_1^t|x^{t-1}) \prod_{j=2}^{n} p_{x_j^t|x_{j-1}^t,x^{t-1}}(x_j^t|x_{j-1}^t, x^{t-1}). \tag{34}$$

Moreover, for $j = 2, \ldots, n-1$ we assume that x_j^t in $p_{x_j^t|x_{j-1}^t,x^{t-1}}(x_j^t|x_{j-1}^t, x^{t-1})$ only depends on (in addition to x_{j-1}^t of the vector x^t) the three elements x_{j-1}^{t-1}, x_j^{t-1} and x_{j+1}^{t-1} of the vector x^{t-1}. Thereby,

$$p_{x_j^t|x_{j-1}^t,x^{t-1}}(x_j^t|x_{j-1}^t, x^{t-1}) = p_{x_j^t|x_{j-1}^t,x_{j-1}^{t-1},x_j^{t-1},x_{j+1}^{t-1}}(x_j^t|x_{j-1}^t, x_{j-1}^{t-1}, x_j^{t-1}, x_{j+1}^{t-1}) \tag{35}$$

for $j = 2, \ldots, n-1$. For $j = 1$ and $j = n$ we correspondingly assume

$$p_{x_1^t|x^{t-1}}(x_1^t|x^{t-1}) = p_{x_1^t|x_1^{t-1},x_2^{t-1}}(x_1^t|x_1^{t-1}, x_2^{t-1}) \tag{36}$$

and

$$p_{x_n^t|x_{n-1}^t,x^{t-1}}(x_n^t|x_{n-1}^t, x^{t-1}) = p_{x_n^t|x_{n-1}^t,x_{n-1}^{t-1},x_n^{t-1}}(x_n^t|x_{n-1}^t, x_{n-1}^{t-1}, x_n^{t-1}). \tag{37}$$

In the following, we first discuss the specification of Eq. (35). To obtain a model where the spatial distribution of sand stone and shale does not change in time we set for all $x_{j-1}^t, x_{j-1}^{t-1}, x_{j+1}^{t-1} \in \{0,1,2\}$,

$$p_{x_j^t|x_{j-1}^t,x_{j-1}^{t-1},x_j^{t-1},x_{j+1}^{t-1}}(x_j^t|x_{j-1}^t,x_{j-1}^{t-1},x_j^{t-1}=2,x_{j+1}^{t-1}) = \begin{cases} 1, & \text{for } x_j^t = 2 \\ 0, & \text{otherwise,} \end{cases} \tag{38}$$

and

$$p_{x_j^t|x_{j-1}^t,x_{j-1}^{t-1},x_j^{t-1},x_{j+1}^{t-1}}(x_j^t=2|x_{j-1}^t,x_{j-1}^{t-1},x_j^{t-1},x_{j+1}^{t-1}) = 0, \qquad \text{for } x_j^{t-1} \in \{0,1\}. \tag{39}$$

For the remaining probabilities in Eq. (35), we adopt the same values as used in Loe and Tjelmeland (2021a), see Table 1. The reasoning behind these probabilities is that if $x_j^{t-1} = 1$ the probability for having $x_j^t = 1$ should be high, and in particular this probability should be high if also $x_{j-1}^t = 1$. If $x_j^{t-1} = 0$ the probability for having also $x_j^t = 0$ should be high unless $x_{j-1}^t = x_{j-1}^{t-1} = x_{j+1}^{t-1} = 1$.

The probabilities in Eqs. (36) and (37) we simply define from the values set for the probabilities in Eq. (35) by defining the values lying outside the simulated lattice to be zero. For x^1 we define that all the elements should be equal to 0 or 2, and assume the elements to be independent with $p_{x_j^1}(x_j^1 = 2) = 1/40$ and $p_{x_j^1}(x_j^1 = 0) = 1 - 1/40$. This results in a vector x^1 with a few (typically one node thick) layers of shale, with the remaining elements being oil filled sand stone. One realisation from the specified Markov process for $\{x^t\}_{t=1}^T$ is shown in Figure 7(a). This realisation is also used to simulate the observations used in the simulation example.

For the likelihood $f_{y^t|x^t}(y^t|x^t)$, we know from Section 4 that it is sufficient to specify $f_{y_j^t|x_j^t}(y_j^t|x_j^t)$ since the elements of y^t are assumed to be conditionally independent given x^t, with y_j^t only depending on x_j^t. To avoid that the likelihood involves an ordering of the three possible values of x_j^t, we let y_j^t be a vector with two components, $y_j^t = (y_{j,1}^t, y_{j,2}^t)$, and choose $f_{y_j^t|x_j^t}(y_j^t|x_j^t)$ as a bivariate Gaussian distribution $\mathcal{N}(y_j^t; \mu(x_j^t), \Sigma)$ with a mean vector

$$\mu(x_j^t) = \begin{cases} (0,0) & \text{if } x_j^t = 0, \\ (1,0) & \text{if } x_j^t = 1, \\ (\frac{1}{2}, \frac{\sqrt{3}}{2}) & \text{if } x_j^t = 2, \end{cases} \tag{40}$$

and covariance matrix $\Sigma = \sigma^2 I$. As illustrated in Figure 8, the mean vectors

Table 1: Simulation experiment: Probabilities defining the true forward model $p_{x^t|x^{t-1}}(x^t|x^{t-1})$ of the Markov process $\{x\}_{t=1}^T$ that are not specified in Eqs. (38) or (39).

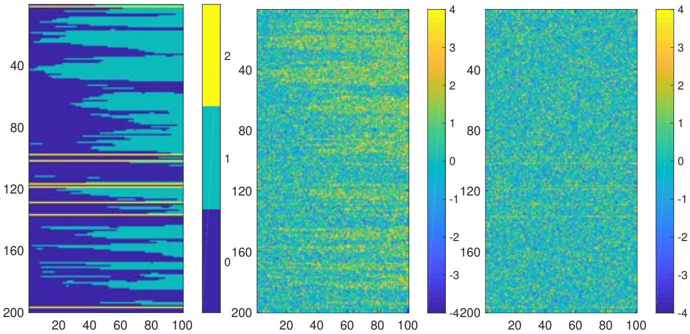
x^t_{j-1}	x^{t-1}_{j-1}	x^{t-1}_{j+1}	$p(x^t_j = 1 \mid x^t_{j-1}, x^{t-1}_{j-1}, x^{t-1}_j = 0, x^{t-1}_{j+1})$	$p(x^t_j = 1 \mid x^t_{j-1}, x^{t-1}_{j-1}, x^{t-1}_j = 1, x^{t-1}_{j+1})$
0	0	0	0.0050	0.9800
0	0	1	0.0400	0.9800
0	0	2	0.0050	0.9800
0	1	0	0.0100	0.9900
0	1	1	0.0400	0.9800
0	1	2	0.0100	0.9800
0	2	0	0.0050	0.9900
0	2	1	0.0400	0.9800
0	2	2	0.0050	0.9800
1	0	0	0.0100	0.9900
1	0	1	0.0400	0.9999
1	0	2	0.0100	0.9999
1	1	0	0.0400	0.9999
1	1	1	0.9800	0.9999
1	1	2	0.0400	0.9999
1	2	0	0.0100	0.9999
1	2	1	0.0400	0.9999
1	2	2	0.0100	0.9999
2	0	0	0.0050	0.9999
2	0	1	0.0400	0.9999
2	0	2	0.0050	0.9999
2	1	0	0.0100	0.9999
2	1	1	0.0400	0.9999
2	1	2	0.0100	0.9999
2	2	0	0.0050	0.9999
2	2	1	0.0400	0.9999
2	2	2	0.0050	0.9999

$\mu(0)$, $\mu(1)$ and $\mu(2)$ are chosen to lie at the vertices of an equilateral triangle with unit sides. This is to avoid an ordering of the three classes. We assume in this simulation experiment that the true likelihood model $p_{y^t|x^t}(y^t|x^t)$ and the assumed likelihood model $f_{y^t|x^t}(y^t|x^t)$ are equal. As such, the assumed likelihood model $f_{y^t|x^t}(y^t|x^t)$ is used to generate the observation process $\{y^t\}_{t=1}^T$. Specifically, using the simulated Markov process shown in Figure 7(a) and setting $\sigma = 1.0$, we generate $\{y^t\}_{t=1}^T$ by simulating, independently for each $j = 1, \dots, 200$ and $t = 1, \dots, 100$,

$$y^t_j \sim f_{y^t_j|x^t_j}(\cdot|x^t_j).$$

An image of $\{(y^t_{j,1}, j = 1, \dots, n)\}_{t=1}^T$ is shown in Figure 7(b) and an image of $\{(y^t_{j,2}, j = 1, \dots, n)\}_{t=1}^T$ is shown in Figure 7(c).

When running the proposed updating procedure, we need to set a value for



(a) **(b)** **(c)**

Figure 7: Simulation experiment: (a) The latent Markov process $\{x^t\}_{t=1}^{100}$, (b) the first coordinates $\{(y_{j,1}^t, j = 1, \dots, 200)\}_{t=1}^{100}$ of the observation process $\{y^t\}_{t=1}^{T}$, and (c) the second coordinates $\{(y_{j,2}^t, j = 1, \dots, 200)\}_{t=1}^{100}$

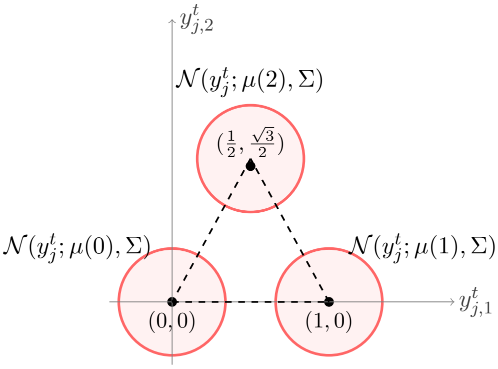

$y_{j,2}^t$
$\mathcal{N}(y_j^t; \mu(2), \Sigma)$
$(\frac{1}{2}, \frac{\sqrt{3}}{2})$
$\mathcal{N}(y_j^t; \mu(0), \Sigma)$
$\mathcal{N}(y_j^t; \mu(1), \Sigma)$
$y_{j,1}^t$
$(0, 0)$
$(1, 0)$

Figure 8: Simulation experiment: Illustration of assumed likelihood model $f_{y_j^t|x_j^t}(y_j^t|x_j^t)$

ν, the order of the assumed Markov chain model $f_{x^t|\theta^t}(x^t|\theta^t)$, and a value for the integer d in Eq. (28) which determines the structure of $q(x^{t,(i)}, z^{t,(i)}; \theta^t, y^t)$. High values for ν and d, and high values for d especially, make the construction of $q(x^{t,(i)}, z^{t,(i)}; \theta^t, y^t)$ computer-demanding. Below, we investigate the two values $\nu = 1$ and $\nu = 2$, and for each of these we consider the three values $d = 1$, $d = 2$ and $d = 3$. Thereby, we have six combinations, or cases, for (ν, d). For each of these six cases, we perform five independent runs, using ensemble size $M = 20$. For each run, an initial ensemble $\{x^{1,(1)}, \dots, x^{1,(M)}\}$ is generated by simulating independent samples from the initial model $p_{x^1}(x^1)$ of the Markov process specified above. The hyper-parameters $a_0^t(0), \dots, a_0^t(K^\nu - 1)$, $a_i^{t,j}(0), \dots, a_i^{t,j}(K-1)$ of the prior distribution $f_{\theta^t}(\theta^t)$ for θ^t (cf. Section A.1 in in Appendix A) at each time

Table 2: Results from simulation experiment: Proportion of correctly classified variables x_j^t obtained with the MAP estimates in Eq. (41) computed in five independent runs

$d=1,\nu=1$	$d=2,\nu=1$	$d=3,\nu=1$	$d=1,\nu=2$	$d=2,\nu=2$	$d=3,\nu=2$
0.8649	0.8903	0.8912	0.8472	0.8831	0.8688

step t are all set equal to one, and 500 iterations are used in the MCMC simulation of $\theta^{t,(i)}|x^{t,-(i)}, y^t$ (cf. Section A.2 in Appendix A).

5.2 Results

To evaluate the performance of the proposed approach, we first compute, for each of the five runs of each of the six combinations of (ν, d), the maximum a posteriori probability (MAP) estimate $\hat{x}_t^j$ of x_j^t, $t = 1, \dots, T$, $j = 1, \dots, n$,

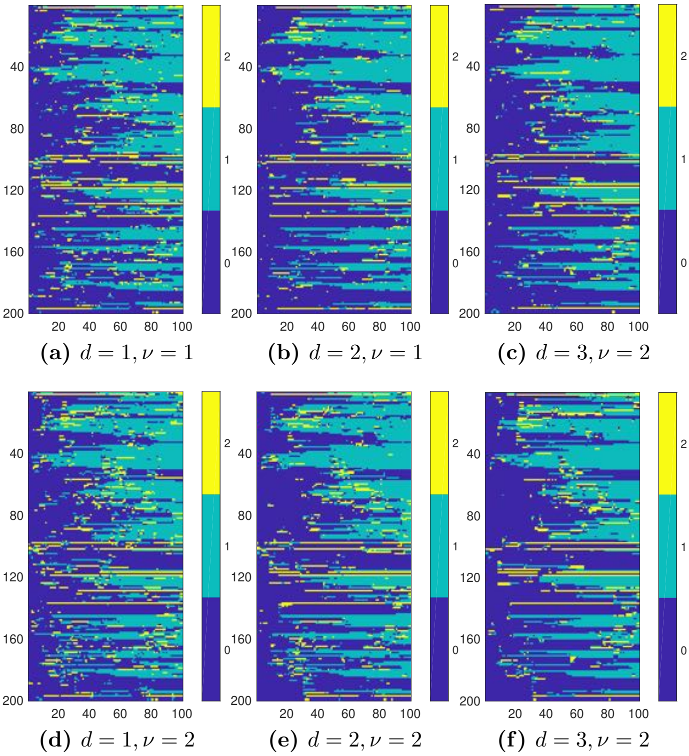
$$\hat{x}_j^t = \underset{k}{\text{argmax}} \left\{ \hat{p}_j^t(k) \right\}, \tag{41}$$

where

$$\hat{p}_j^t(k) = \frac{1}{M} \sum_{i=1}^{M} 1(z_j^{t,(i)} = k), \quad k = 0, 1, 2, \tag{42}$$

is an estimate of the marginal filtering probability $p_{x_j^t|y^{1:t}}(k|y^{1:t})$. Figure 9 shows images of the computed MAP estimates $\{\hat{x}_j^t, j = 1, \dots, n\}_{t=1}^T$ from one of the five runs performed for each of the six cases. From a visual inspection, it seems that we in all cases manage to capture the main characteristics of the true x^t-process in Figure 7(a), but the MAPs shown in Figures 9(a) and (d), which are obtained using $d = 1$, are possibly a bit noisier than the others. Table 2 lists the ratio of correctly classified variables x_j^t based on the MAPs obtained from the five independent runs of each case. According to Table 2, we classify around 85-90% of the variables correctly, and the best results are obtained when using the combinations $\nu = 1, d = 2$ and $\nu = 1, d = 3$, i.e. when adopting a first-order Markov chain ($\nu = 1$) for $f_{x^t|\theta^t}(x^t|\theta^t)$ and using 2×2- or 2×3-cliques ($d = 2$ or $d = 3$) in the construction of $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$.

To further investigate the performance of the proposed approach, we estimate for each j and t the probability that $z_j^{t,(i)}$ is equal to the true value x_j^t, and we do this for each of the classes $k = 0, 1, 2$. Specifically, for each run and for each



(a) $d = 1, \nu = 1$ **(b)** $d = 2, \nu = 1$ **(c)** $d = 3, \nu = 2$

(d) $d = 1, \nu = 2$ **(e)** $d = 2, \nu = 2$ **(f)** $d = 3, \nu = 2$

Figure 9: Results from simulation experiment: MAP estimates of $\{x_j^t, j = 1, \ldots, 200\}_{t=1}^{100}$

$j = 1, \ldots, n$, $t = 1, \ldots, T$, we compute, if $x_j^t = 0$,

$$\pi_{0|0} = \frac{1}{M} \sum_{i=1}^{M} 1(z_j^{t,(i)} = 0),$$

while if $x_j^t = 1$, we compute

$$\pi_{1|1} = \frac{1}{M} \sum_{i=1}^{M} 1(z_j^{t,(i)} = 1),$$

and if $x_j^t = 2$, we compute

$$\pi_{2|2} = \frac{1}{M} \sum_{i=1}^{M} 1(z_j^{t,(i)} = 2).$$

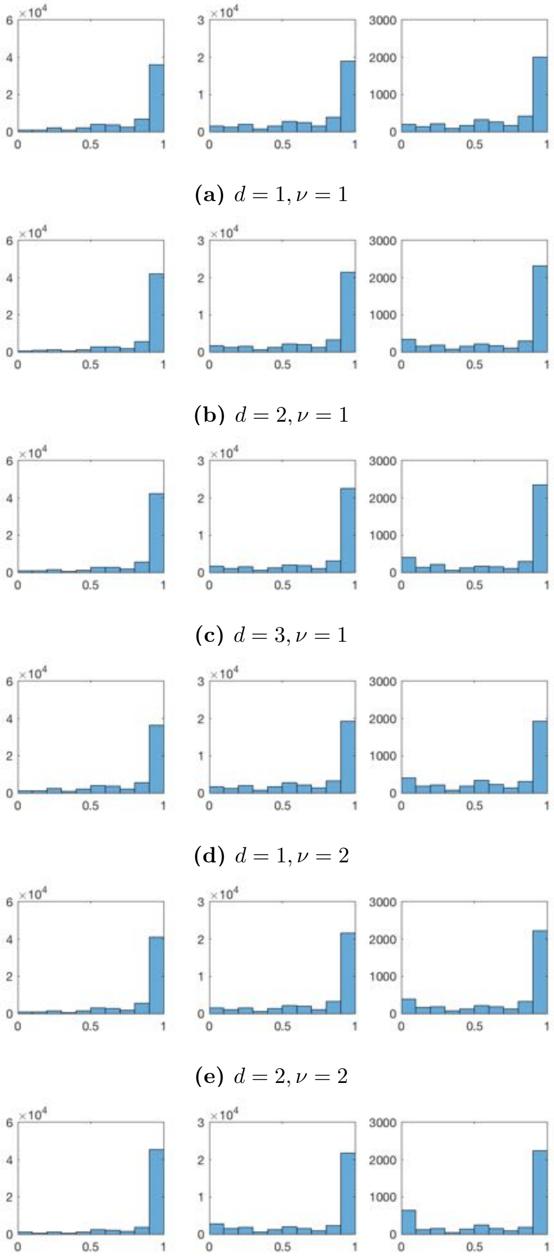
Table 3: Results from simulation experiment: Estimated probabilities for observing $z_j^{t,(i)}$ equal to the true value x_j^t for each class $k = 0, 1, 2$

	$d = 1, \nu = 1$	$d = 2, \nu = 1$	$d = 3, \nu = 1$	$d = 1, \nu = 2$	$d = 2, \nu = 2$	$d = 3, \nu = 2$
$\bar{\pi}_{0\|0}$	0.8210	0.8685	0.8687	0.8151	0.8587	0.8848
$\bar{\pi}_{1\|1}$	0.7558	0.7837	0.7964	0.7508	0.7840	0.7590
$\bar{\pi}_{2\|2}$	0.7423	0.7480	0.7412	0.6935	0.7285	0.6985
$\bar{\pi}$	0.7730	0.8001	0.8021	0.7531	0.7904	0.7808

There are, in the latent x^t-process shown in Figure 7(a), 11929 variables x_j^t taking the value 0, 7271 variables taking the value 1 and 800 variables taking the value 2. Thereby, since we run each of the six (ν, d) combinations five times, we obtain for each (ν, d) combination $5 \cdot 11929$ samples of $\pi_{0|0}$, $5 \cdot 7271$ samples of $\pi_{1|1}$ and $5 \cdot 800$ samples of $\pi_{2|2}$. We denote the corresponding sample means by $\bar{\pi}_{0|0}$, $\bar{\pi}_{1|1}$ and $\bar{\pi}_{2|2}$, and we let $\bar{\pi} = \frac{1}{3}\left(\bar{\pi}_{0|0} + \bar{\pi}_{1|1} + \bar{\pi}_{2|2}\right)$. Figure 10 presents histograms constructed from the samples of $\pi_{0|0}$, $\pi_{1|1}$ and $\pi_{2|2}$ for each case, and Table 3 summarises the corresponding computed values for $\bar{\pi}_{0|0}$, $\bar{\pi}_{1|1}$, $\bar{\pi}_{2|2}$ and $\bar{\pi}$. The values for $\bar{\pi}$ indicate that, again, we obtain the best results using $\nu = 1, d = 2$ and $\nu = 1, d = 3$. Computationally, using $d = 3$ is more demanding, and since the improvement it offers over $d = 2$ is only minor, the best approach may be to use $\nu = 1, d = 2$.

6 Closing remarks

An ensemble updating method for categorical state vectors is proposed. The proposed procedure is an improved version of the updating procedure for categorical vectors described in Loe and Tjelmeland (2021a). What is new is mainly in how the optimal solution of $q(x^{t,(i)}, z^{t,(i)}; \theta^t, y^t)$ is computed. Loe and Tjelmeland (2021a) construct the conditional distribution $q(z^{t,(i)}|x^{t,(i)}; \theta^t, y^t)$ directly based on a directed acyclic graph (DAG) for the dependency properties of $q(x^{t,(i)}, z^{t,(i)}; \theta^t, y^t)$. The chosen structure of the DAG allows the optimal solution of $q(z^{t,(i)}|x^{t,(i)}; \theta^t, y^t)$ to be computed recursively using a combination of dynamic and linear programming. This strategy works well when the elements of x^t are binary, but the algorithm is difficult to generalise to situations with more than two classes. Moreover, $f_{x^t|\theta^t}(x^t|\theta^t)$ is restricted to be a first-order Markov chain, and it is difficult, or essentially impossible, to generalise the algorithm to allow for more complicated models with higher-order interactions. In the present article, we start with an undirected, decomposable graph instead of a DAG, and the re-



(a) $d = 1, \nu = 1$

(b) $d = 2, \nu = 1$

(c) $d = 3, \nu = 1$

(d) $d = 1, \nu = 2$

(e) $d = 2, \nu = 2$

(f) $d = 3, \nu = 2$

Figure 10: Results from the simulation experiment: Histograms of $\pi_{0|0}$ (left), $\pi_{1|1}$ (middle) and $\pi_{2|2}$ (right)

sult is a more flexible and efficient updating procedure. The proposed procedure is demonstrated in a simulation example with three classes, and the results look promising.

In Section 3.2, we introduced an exact and an approximate class of distributions for the updating of a forecast ensemble member $x^{t,(i)}$. Although it may seem disadvantageous to pursue an approximate approach over an exact one, we believe that in this case the approximate approach actually provides better results. The constraints of the approximate approach are less restrictive and allows the optimality criterion to affect the solution to a larger extent. This may in turn result in an optimal updating distribution which is more robust against the assumptions of the assumed Bayesian model. That is, even if the assumed Markov chain models $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t)$ are far from the truth, the optimal updating distribution $q(x^{t,(i)}, z^{t,(i)}; \theta^{t,(i)}, y^t)$ may still provide reasonably good results.

Future work naturally includes to extend the proposed procedure to two dimensions. Assuming x^t is defined on a two-dimensional grid, a possible choice of model for $f_{x^t|\theta^t}(x^t|\theta^t)$ is then a Markov mesh model (Abend et al., 1965). However, the two-dimensional situation makes it more difficult to construct $q(x^{t,(i)}, z^{t,(i)}; \theta^t, y^t)$, and it is probably necessary to introduce some sort of approximations to overcome these difficulties.

References

Abend, K., Harley, T., & Kanal, L. (1965). Classification of binary random patterns. *IEEE Transactions on Information Theory*, *11*, 538–544.

Bishop, C. H., Etherton, B. J., & Majumdar, S. J. (2001). Adaptive sampling with the ensemble transform Kalman filter. Part 1: Theoretical aspects. *Monthly Weather Review*, *129*, 420–436.

Burgers, G., van Leeuwen, P. J., & Evensen, G. (1998). Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, *126*, 1719–1724.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spiegelhalter, D. J. (1999). *Probabilistic networks and expert systems: Exact computational methods for bayesian networks*. Springer-Verlag New York.

Cressie, N. A. (1993). *Statistics for spatial data*. Wiley, New York.

Doucet, A., de Freitas, N., & Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. Springer-Verlag, New York.

Evensen, G. (2003). The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, *53*, 343–367.

Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEE-Proceedings-F*, *140*, 107–113.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, *82*, 35–45.

Kindermann, R., & Snell, L. (1980). *Markov random fields and their applications.* American Mathematical Society.

Künsch, H. R. (2000). State space and hidden Markov models. In O. E. Barndorff-Nielsen, D. R. Cox, & C. Klüppelberg (Eds.), *Complex stochastic systems.* Chapman and Hall/CRC, Chap. 3, p. 109-174.

Loe, M. K., & Tjelmeland, H. (2021a). A generalised and fully Bayesian framework for ensemble updating [arXiv:2103.14565 [stat.ME]].

Loe, M. K., & Tjelmeland, H. (2021b). Ensemble updating of binary state vectors by maximising the expected number of unchanged components [DOI: 10.1111/sjos.12483]. *Scandinavian Journal of Statistics,* To Appear.

Snyder, C., Bengtsson, T., Bickel, P., & Anderson, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, *136*, 4629–4640.

Tippett, M. K., Anderson, J. L., Bishop, C. H., & Hamill, T. M. (2003). Ensemble square root filters. *Monthly Weather Review*, *131*, 1485–1490.

Whitaker, J. S., & Hamill, T. M. (2002). Ensemble data assimilation without perturbed observations. *Monthly Weather Review*, *130*, 1913–1924.

A Appendix

This appendix provides some additional details about the specification of $f_{\theta^t}(\theta^t)$ for the assumed Bayesian model in Section 4.1, and explains how to simulate a realisation from the distribution $f_{\theta^t|x^{t,-(i)},y^t}(\theta^t|x^{t,-(i)},y^t)$.

A.1 Parameter specification

Here, we specify the distribution $f_{\theta^t}(\theta^t)$ of Section 4.1 in more detail. Recall from Section 4.1 that $f_{x^t|\theta^t}(x^t|\theta^t)$ is a Markov chain of order ν and that θ^t is a vector, $\theta^t = (\theta_0^t, \theta_1^t, \ldots, \theta_{n-\nu+1}^t)$, where θ_0^t represents the probabilities of $f_{x_{1:\nu}^t|\theta^t}(x_{1:\nu}^t|\theta^t)$ and θ_i^t, $i = 1, \ldots, n-\nu$, represents the $K^\nu \times K$ transition matrix

$f_{x^t_{i+\nu}|x^t_{i:i+\nu-1},\theta^t}(x^t_{i+\nu}|x^t_{i:i+\nu-1},\theta^t)$. To simplify some of the following notations, we will make use of the notation

$$N(v) = \sum_{j=1}^{V} K^{V-j} v_j$$

where $v = (v_1, \ldots, v_V)$ is a vector of V categorical variables $v_j \in \{0, 1, \ldots, K-1\}$. Each configuration of the vector v thereby corresponds to an integer $N(v) \in \{0, \ldots, K^V - 1\}$. Now, let

$$\theta^t_0 = (\theta^t_0(0), \theta^t_0(1), \ldots, \theta^t_0(K^\nu - 1))$$

and

$$\theta^t_0(N(x^t_{1:\nu})) = f_{x^t_{1:\nu}|\theta^t}(x^t_{1:\nu|\theta^t}).$$

Hence, if for example $f_{x^t|\theta^t}(x^t|\theta^t)$ is a Markov chain of order $\nu = 3$, then $\theta^t_0(N(0,0,0)) = \theta^t_0(0)$ is the probability that $(x^t_1, x^t_2, x^t_3) = (0,0,0)$, while $\theta^t_0(N(0,0,1)) = \theta^t_0(1)$ is the probability that $(x^t_1, x^t_2, x^t_3) = (0,0,1)$. Next, let

$$\theta^t_i = (\theta^{t,0}_i, \ldots, \theta^{t,K^\nu-1}_i)^T \qquad \text{and} \qquad \theta^{t,j}_i = (\theta^{t,j}_i(0), \ldots, \theta^{t,j}_i(K-1))$$

so that $\theta^{t,j}_i$, $j = 0, \ldots, K^\nu - 1$ represents row number $j+1$ of the transition matrix θ^t_i, and

$$\theta^{t,N(x_{i:i+\nu-1})}_i(x^t_{i+\nu}) = f_{x^t_{i+\nu}|x^t_{i:i+\nu-1},\theta^t}(x^t_{i+\nu}|x^t_{i:i+\nu-1},\theta^t).$$

Hence, if for example $\nu = 3$, then $\theta^{t,N(0,0,0)}_1(0)$ is the probability that $x^t_4 = 0$ given that $(x^t_1, x^t_2, x^t_3) = (0,0,0)$, while $\theta^{t,N(0,0,1)}_1(0)$ is the probability that $x^t_4 = 0$ given that $(x^t_1, x^t_2, x^t_3) = (0,0,1)$. To obtain a prior $f_{\theta^t}(\theta^t)$ which is conjugate for $f_{x^t|\theta^t}(x^t|\theta^t)$ when $f_{x^t|\theta^t}(x^t|\theta^t)$ is a Markov chain, we start by assuming that θ^t_0, $\theta^{t,0}_1$, $\theta^{t,1}_1$, …, $\theta^{t,K^\nu-1}_{n-\nu}$ are all independent a priori,

$$f_{\theta^t}(\theta^t) = f_{\theta^t_0}(\theta^t_0) \prod_{i=1}^{n-\nu} \prod_{j=0}^{K^\nu-1} f_{\theta^{t,j}_i}(\theta^{t,j}_i).$$

Next, we adopt a Dirichlet distribution for each of the vectors θ^t_0, $\theta^{t,0}_1$, $\theta^{t,1}_1$, …, $\theta^{t,K^\nu-1}_{n-\nu}$. Specifically, we adopt for θ^t_0 a Dirichlet distribution with known hyperparameters $a^t_0(0), \ldots, a^t_0(K-1)$, and for each $\theta^{t,j}_i$ we adopt a Dirichlet distribution with known hyper-parameters $a_i^{t,j}(0), \ldots, a_i^{t,j}(K-1)$. Then,

$$f_{\theta_0^t}(\theta_0^t) \propto \prod_{k=0}^{K-1} \left(\theta_0^t(k)\right)^{a_0^t(k)}$$

and

$$f_{\theta_i^{t,j}}(\theta_i^{t,j}) \propto \prod_{k=0}^{K-1} \left(\theta_i^{t,j}(k)\right)^{a_i^{t,j}(k)}.$$

A.2 Parameter simulation

A necessary step of the proposed ensemble updating procedure is to simulate a parameter $\theta^{t,(i)}|x^{t,-(i)}, y^t$. Generally, for the assumed Bayesian model introduced in Section 3.1, this can be achieved by introducing x^t as an auxiliary variable and construct a Gibbs sampler which simulates (x^t, θ^t) from the joint distribution

$$f_{x^t,\theta^t|x^{t,-(i)},y^t}(x^t, \theta^t|x^{t,-(i)}, y^t) \propto f_{\theta^t}(\theta^t) f_{x^t|\theta^t}(x^t|\theta^t) f_{y^t|x^t}(y^t|x^t) \prod_{j\neq i} f_{x^t|\theta^t}(x^{t,(j)}|\theta^t).$$

The Gibbs sampler alternates between drawing x^t and θ^t from the full conditional distributions $f_{x^t|\theta^t,x^{t,-(i)},y^t}(x^t|\ \theta^t, x^{t,-(i)}, y^t)$ and $f_{\theta^t|x^t,x^{t,-(i)},y^t}(\theta^t|x^t, x^{t,-(i)}, y^t)$, respectively. From the dependency assumptions of the assumed Bayesian model (see Figure 4), it follows that

$$f_{x^t|\theta^t,x^{t,-(i)},y^t}(x^t|\theta^t, x^{t,-(i)}, y^t) = f_{x^t|\theta^t,y^t}(x^t|\theta^t, y^t) \tag{43}$$

and

$$f_{\theta^t|x^t,x^{t,-(i)},y^t}(\theta^t|x^t, x^{t,-(i)}, y^t) = f_{\theta^t|x^t,x^{t,-(i)}}(\theta^t|x^t, x^{t,-(i)}). \tag{44}$$

Both of these distributions are tractable, so the Gibbs sampler can be implemented without complications.

Suppose now that $f_{x^t|\theta^t}(x^t|\theta^t)$ and $f_{y^t|x^t}(y^t|x^t)$ are as specified in Section 4.1, and that $f_{\theta^t}(\theta^t)$ is as specified in Section A.1 above. The distribution in Eq. (43) then becomes a Markov chain whose initial and transition probabilities can be computed with a forward-backward recursive procedure. For the distribution in Eq. (44), it can easily be shown that θ_0^t, $\theta_j^{t,k}$, $j = 1, \ldots, n-\nu$, $k = 0, \ldots, K^\nu - 1$ are all independent given x^t and $x^{t,-(i)}$, i.e.

$$f_{\theta^t|x^t,x^{t,-(i)}}(\theta^t|x^t, x^{t,-(i)}) = f_{\theta_0^t|x^t,x^{t,-(i)}}(\theta_0^t|x^t, x^{t,-(i)}) \prod_{j,k} f_{\theta_j^{t,k}|x^t,x^{t,-(i)}}(\theta_j^{t,k}|x^t, x^{t,-(i)}).$$

Moreover, $\theta_0^t | x^t, x^{t,-(i)}$ is Dirichlet distributed with parameters

$$\tilde{a}_0^t(r) = a_0^t(r) + 1\left(N(x_{1:\nu}^t) = r\right) + \sum_{m \neq i} 1\left(N(x_{1:\nu}^{t,(m)}) = r\right),$$

for $r = 0, \ldots, K^\nu - 1$, and each $\theta_j^{t,k} | x^t, x^{t,-(i)}$, for $j = 1, \ldots, n - \nu$ and $k = 0, \ldots, K^\nu - 1$, is Dirichlet distributed with parameters

$$\tilde{a}_j^{t,k}(r) = a_j^{t,k}(r) + 1\left(N(x_{j:j+\nu-1}^t) = k\right) 1\left(x_{\nu+j}^t = r\right) + \sum_{m \neq i} 1\left(N(x_{j:j+\nu-1}^{t,(m)}) = k\right) 1\left(x_{\nu+j}^{t,(m)} = r\right),$$

for $r = 0, \ldots, K - 1$.

ISBN 978-82-326-6950-9 (printed ver.)
ISBN 978-82-326-5252-5 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology