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

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.
© 2020 The Authors. *Scandinavian Journal of Statistics* published by John Wiley & Sons Ltd on behalf of The Board of the Foundation of the Scandinavian Journal of Statistics.

## 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.

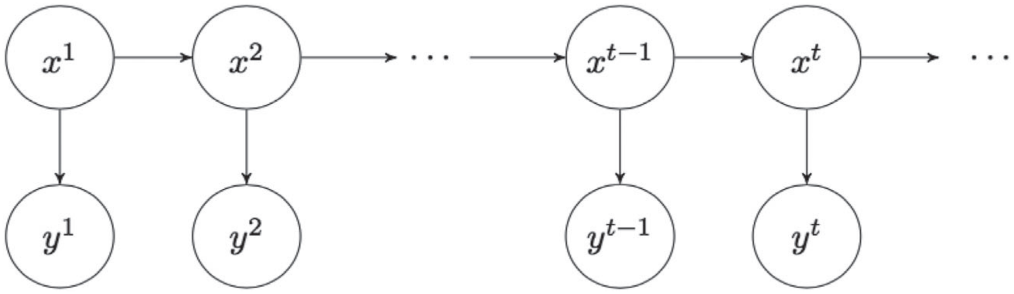


**FIGURE 1** Graphical illustration of the state-space model behind the filtering problem

## 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 $\Omega_X$, while each observation $y^t$ is $k$-dimensional and can take on values in a state space $\Omega_Y$. 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, $y^{1:t} = (y^1, \ldots, y^t)$. 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 $\Omega_X$ consists of a finite number of states, and the integrals in (1a) and (1b) reduce to finite sums. If the number of states in $\Omega_X$ 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:

$$x^1 \sim \mathcal{N}_n(x^1|\mu^1, \Sigma^1),$$

$$x^t|x^{t-1} = A^t x^{t-1} + \omega^t, \quad \omega^t \sim \mathcal{N}_n(\omega|0, \Sigma^t),$$

$$y^t|x^t = H^t x^t + \epsilon^t, \quad \epsilon^t \sim \mathcal{N}_k(\epsilon|0, R^t), \tag{2}$$

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 $\mu$ and analytically tractable covariance $\Sigma$. 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 $(x, \epsilon)$ 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 $\mu$ and $\Sigma$, 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

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

where the subscript $\pi$ 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

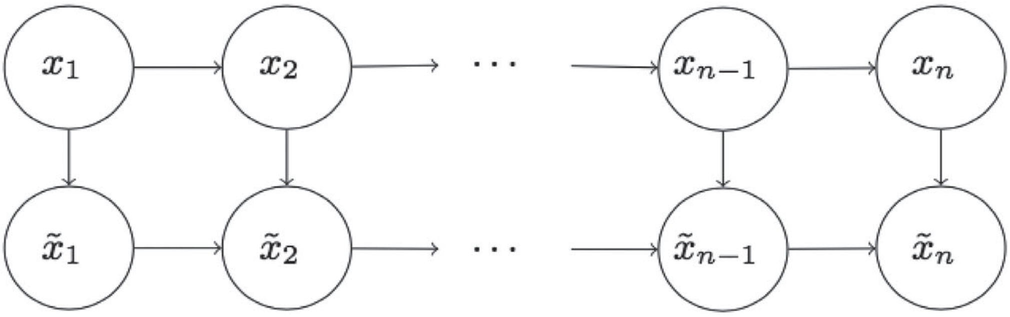



**FIGURE 2** Graphical illustration of the updating distribution $q(\tilde{x}|x, y)$

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 $\pi(\tilde{x}_{i:j}, x_{k:l}|y)$, $1 \le i \le j \le n$, $1 \le k \le l \le n$, 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:

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

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

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 $\pi_k$. Essentially, each $\pi_k$, $k \geq 2$, 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 $\pi_k$. 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 $\pi_k$ 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

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

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

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 $\pi_k$, an essential feature of each $\pi_k$, for $k \geq 2$, is its dependence on $\pi_{k-1}$ 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 $\pi_k$, 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 $\pi_{k-1}$ 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 $t_k$:

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

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

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

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

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

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

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

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],

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

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

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

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:

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

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

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

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

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

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

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

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

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

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

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

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

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

## 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

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

and marginal distributions

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

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

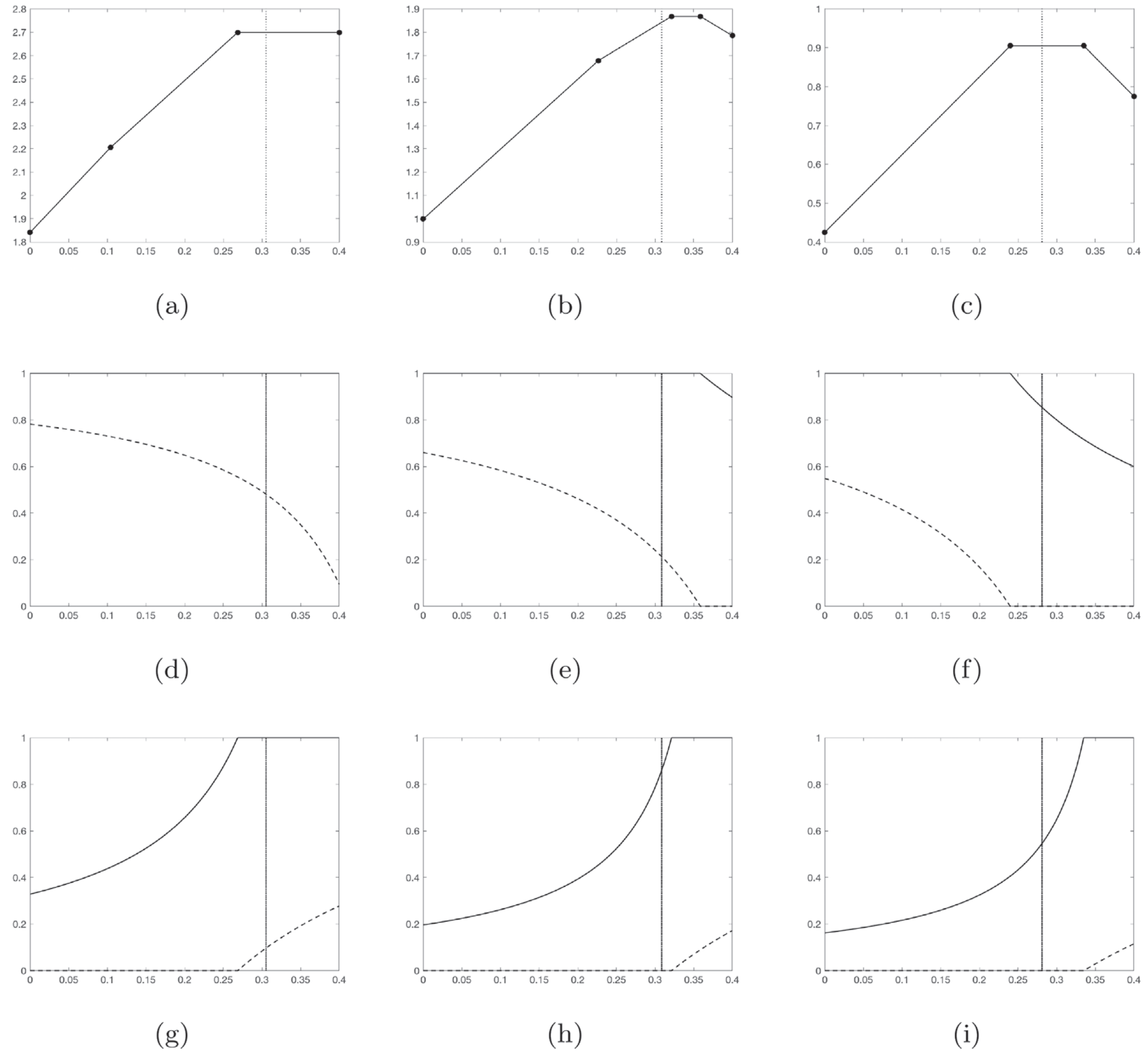



**FIGURE 3** Results from the toy example of Section 5.1. (a–c) $E^*_{k:4}(t_k)$ for $k = 2,3$, and 4, respectively, with the breakpoints highlighted as black dots. (d–f) $q_k^{*00}(t_k)$ (solid) and $q_k^{*01}(t_k)$ (dashed) for $k = 2,3$, and 4, respectively. (g–i) $q_k^{*10}(t_k)$ (solid) and $q_k^{*11}(t_k)$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

| $x_{i-1}^{t-1}$ | $x_i^{t-1}$ | $x_{i+1}^{t-1}$ | $p(x_i^t = 1|x_{i-1}^t = 1, x_{i-1:i+1}^{t-1})$ | $p(x_i^t = 1|x_{i-1}^t = 0, x_{i-1:i+1}^{t-1})$ |
|---|---|---|---|---|
| 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 |

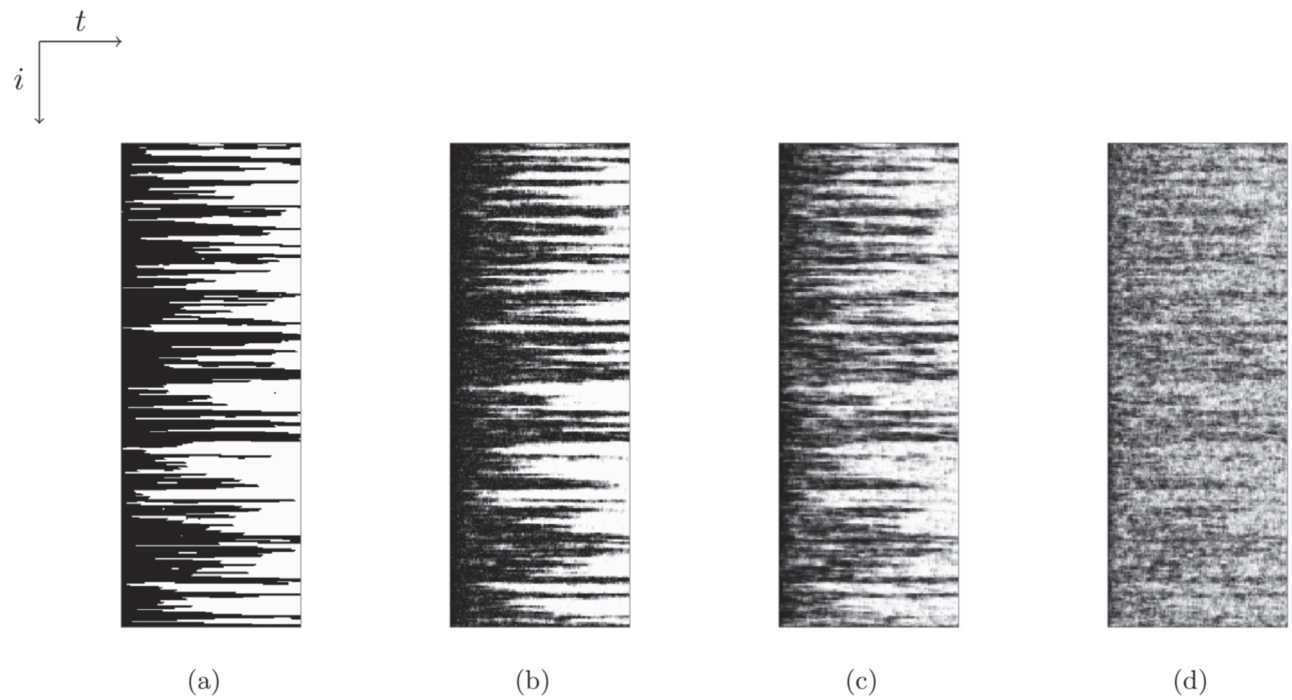



**FIGURE 4** Results from the simulation experiment of Section 5.2: Grayscale images of (a) the unobserved process $\{x^t\}_{t=1}$, (b) $\{\hat{p}_c(x_i^t|y^{1:t})\}_{t=1}^{100}$, (c) $\{\hat{p}_q(x_i^t|y^{1:t})\}_{t=1}^{100}$, and (d) $\{\hat{p}_a(x_i^t|y^{1:t})\}_{t=1}^{100}$. The colors black and 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 $\{(i,t):i = 1, \ldots, n;t = 1, \ldots, T\}$ 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 $y^t = (y_1^t, \ldots, y_n^t)$ to be conditionally independent given $x^t$. Furthermore, we let $y_i^t$ be

Gaussian with mean $x_i^t$ and variance $\sigma^2$. 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 $\sigma^2$ 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 $\sigma^2$ 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 $\theta^t$ represent a vector containing all the parameters required to specify the model $f(x^t|y^{1:t-1})$, we put a prior on $\theta^t$, $f(\theta^t)$, and then set the final estimator for $\theta^t$ 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 $\theta^t$ 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

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

is equal to one.

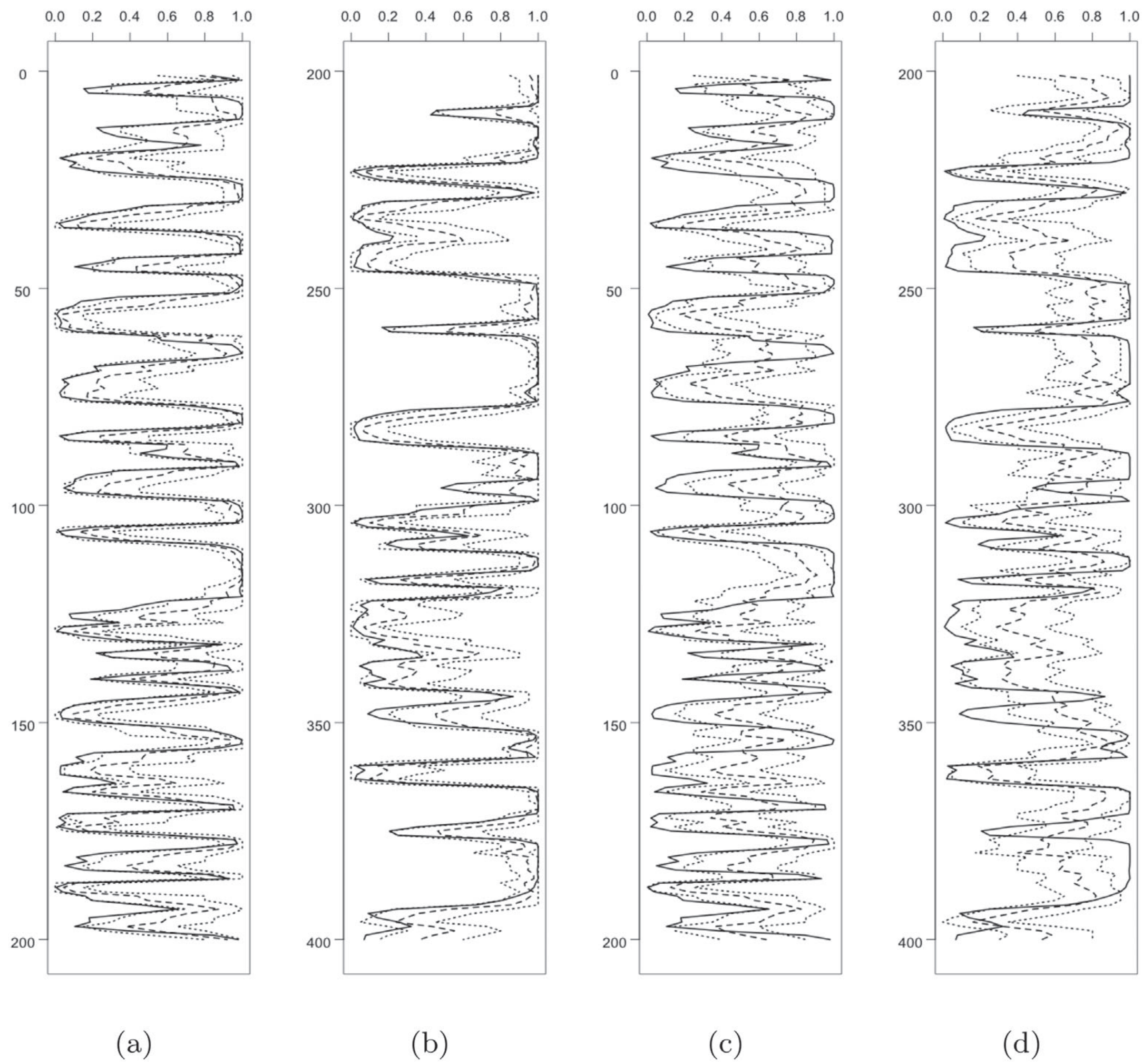



**FIGURE 5** Results obtained at time step $t = 60$ in the numerical experiment of Section 5.2. (a,b) Marginal estimates $\hat{p}_q(x_i^t = 1|y^{1:t})$ (dashed) and corresponding 90% quantile intervals (dotted), in (a) from $i = 1$ to $i = 200$, and in (b) from $i = 201$ to $i = 400$. (c,d) Corresponding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

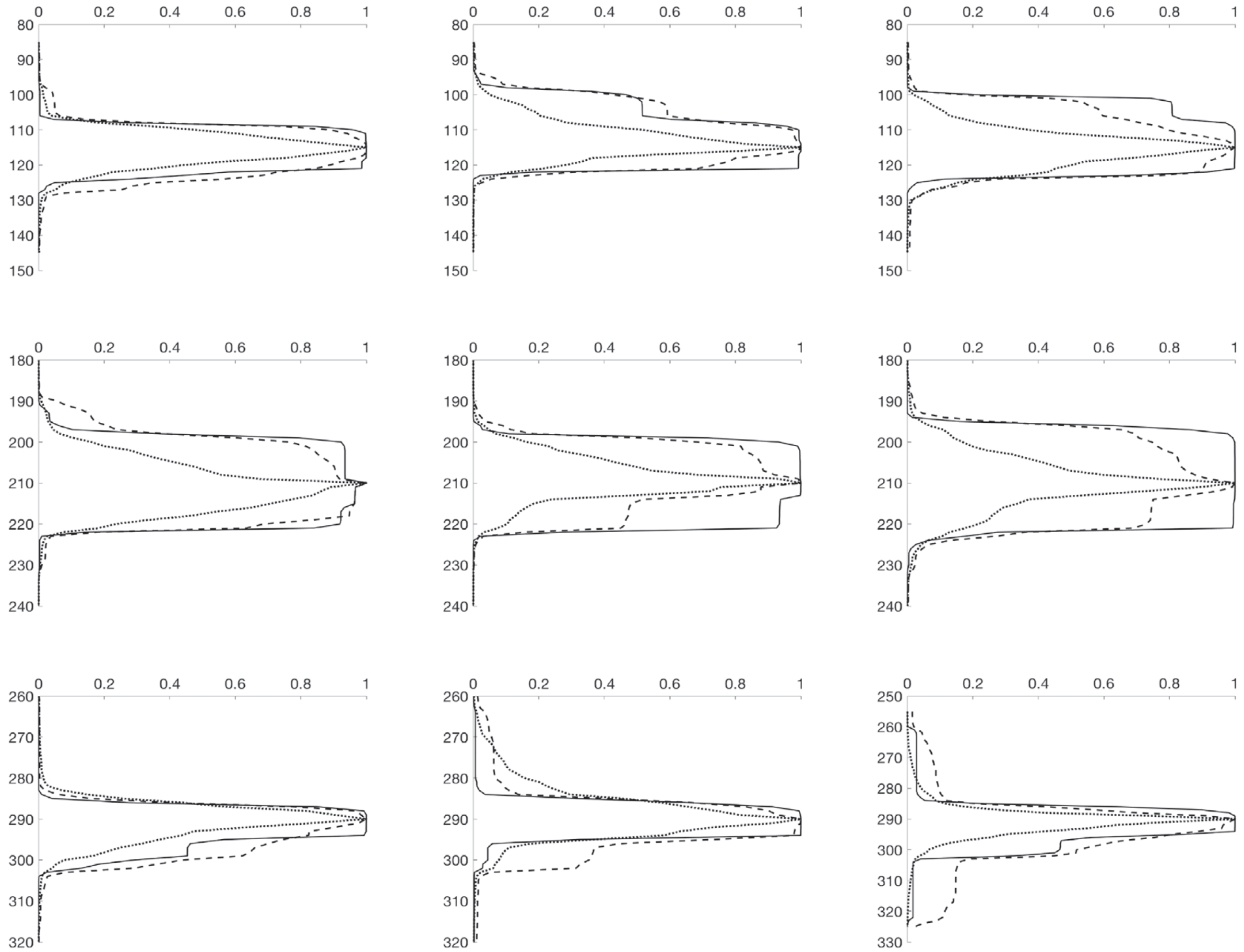


**FIGURE 6** Results from the numerical experiment of Section 5.2. The graphs present $\hat{p}_c^t(i,j)$ (solid), $\hat{p}_q(i,j)$ (dashed), and $\hat{p}_a(i,j)$ (dotted) for the three components $i = 115$, $i = 210$, and $i = 290$ at time steps $t = 60$ (left column), $t = 70$ (middle column), and $t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

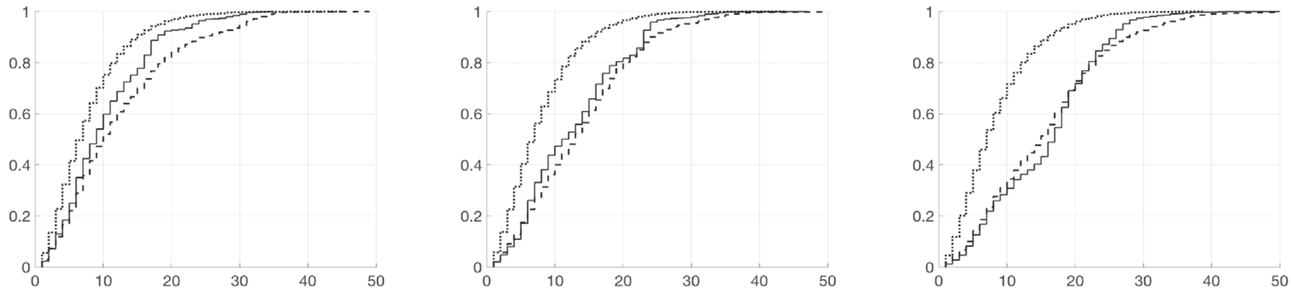
In our second test, we focus on the total number of nodes an arbitrary node $i$ with $x_i^t = 1$ is in contact with. We denote this quantity by $L_i(x^t)$. Mathematically, $L_i(x^t)$ can be written

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

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



(a) (b) (c)

**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 $f(x_i, x_{i+1}|y)$, $i = 1, \ldots\ n-1$. 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

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

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

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 $\pi_n^{00}(t_n) \leq f_n^{00}$, 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 $\pi_n^{10}(t_n) \leq f_n^{10}$, 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 $j$th 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 $q_k^{00}$ and $q_k^{10}$. 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 $q_k^{00}$ and $q_k^{10}$ as

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

where

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

and

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

The corresponding constraints for $q_k^{00}$ and $q_k^{10}$ read

$$\max\left\{0, \frac{f_k^{00} - \pi_k^{01}(t_k)}{\pi_k^{00}(t_k)}\right\} \le q_k^{00} \le \min\left\{1, \frac{f_k^{00}}{\pi_k^{00}(t_k)}\right\}, \tag{A7}$$

$$\max\left\{0, \frac{f_k^{10} - \pi_k^{11}(t_k)}{\pi_k^{10}(t_k)}\right\} \le q_k^{10} \le \min\left\{1, \frac{f_k^{10}}{\pi_k^{10}(t_k)}\right\}, \tag{A8}$$

and

$$t_{k+1}^{B(j)} \le (\rho_k^{0|0} - \rho_k^{0|1})\pi_k^{00}(t_k) q_k^{00} + (\rho_k^{0|0} - \rho_k^{0|1})\pi_k^{10}(t_k) q_k^{10} + (f_k^{00} + f_k^{10})\rho_k^{0|1} \le t_{k+1}^{B(j+1)}. \tag{A9}$$

If for some fixed $t_k \in [t_k^{\min}, t_k^{\max}]$ we consider a coordinate system with $q_k^{00}$ along the first axis and $q_k^{10}$ 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 $q_k^{00}$-direction cf. (A7), two lines in the $q_k^{10}$-direction cf. (A8), and two parallel lines with a negative slope of $-\pi_k^{00}(t_k)/\pi_k^{10}(t_k)$ 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 $q_k^{00}$ and $q_k^{10}$ 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 $q_k^{00}$ and $q_k^{10}$ are as small as possible. For simplicity, we assume in the following that the feasible region is nonempty.

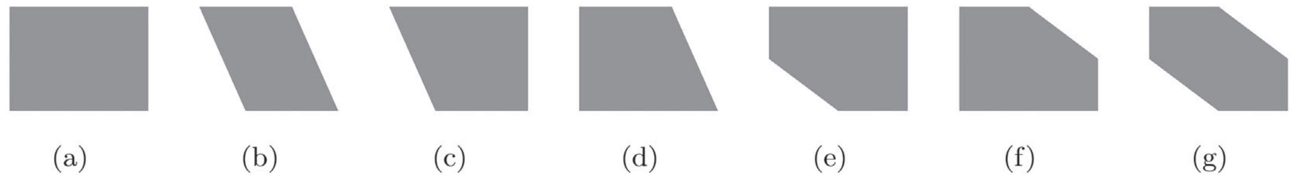



**FIGURE 8** Illustrations of some possible shapes for the feasible regions of the linear programs at the intermediate steps of the backward recursion. The polygons are drawn in a coordinate system with $q_k^{00}$ in the horizontal direction and $q_k^{10}$ in the vertical direction

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 $\delta t_k$ 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 $\delta t_k$ 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.