NTNU
Norwegian University of
Science and Technology

Precipitation forecasting using Radar Data

Tore Botnen

Master of Science in Physics and Mathematics
Submission date: June 2009
Supervisor: Jo Eidsvik, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Problem Description

The main task of this paper is to filter out noise from a series of radar images and to carry out short term precipitation forecasts.

Assignment given: 19. January 2009
Supervisor: Jo Eidsvik, MATH

Preface

In my master thesis TMA4905 which concludes my studies at the Norwegian University of Science and Technology (NTNU), I have studied how to forecast precipitation using a sequence of radar images. The task proved to be challenging but also very rewarding, testing my mathematical skills, my ability to be innovative and has provided me the chance to use my theoretical foundation to solve a practical problem.

The task was first presented to me by my supervisor, associate professor Jo Eidsvik, who in turn had a dialog with SINTEF Energy, Trondheim. The work was done in the spring semester of 2009, from January to June. The preferred programming language is MATLAB, because of its versatile statistical tool box and excellent graphical visualizations.

I would like to thank my supervisor Jo Eidsvik who has been very helpful during this period, providing guidance and relevant information. I would also like to express my gratitude to Lena Tøfte from SINTEF, who provided me with real radar material from the Rissa radar, located in Sør Trøndelag.

Tore Botnen
Trondheim
15.06.2009

Abstract

The main task of this assignment is to filter out noise from a series of radar images and to carry out short term precipitation forecasts. It is important that the final routine is performed online, yielding new forecasts as radar images arrive with time. The data available is a time series arriving at a one hour ratio, from the Rissa radar located in Sør Trøndelag.

Gaussian radial basis functions are introduced to create the precipitation field, whose movement is solely governed by its velocity field, called advection. By performing discretization forward in time, from the basis given by the differential advection equation, prior distributions can be obtained for both basis functions and advection. Assuming normal distributed radar errors, the basis functions and advection are conditioned on associating radar images, which in turn can be taken into the prior distributions, yielding new forecasts. A modification to the model, labeling the basis functions either active or inactive, enable the process of birth and death of new rain showers. The preferred filtering technique is a joint MCMC sampler, but we make some approximations, sampling from a single MCMC sampler, to successfully implement an online routine.

The model yield good results on synthetic data. In the real data situation the filtered images are satisfying, and the forecast images are approximately predicting the forthcoming precipitation.

The model removes statistical noise efficiently and obtain satisfying predictions. However, due to the approximation in the MCMC algorithm used, the variance is somewhat underestimated. With some further work with the MCMC update scheme, and given a higher frequency of incoming data, it is the authors belief that the model can be a very useful tool in short term precipitation forecasting. Using gauge data to estimate the radar errors, and merging online gauge data with incoming radar images using block-Kriging, will further improve the estimates.

Contents

1 Introduction

For thousands of years man has been trying to predict the weather so that he can adjust his nearby future in the best possible manner. Whether it is the farmer hoping for a good harvest, the fisherman going to sea or the businessman considering taking his bicycle to work, they all want to plan ahead using the knowledge of the forthcoming weather as the major factor for making a wise decision. Ancient local knowledge, the use of barometers and in recent years advanced numerical models, have all been important tools in weather forecasting. In this report however, the aim is to use radar images in forecasting and estimating the precipitation.

In many situations there is also an economical aspect to precipitation forecasting. Power companies which produces hydroelectric power, fit into this category. A costly consequence of not having good estimates of the amount of water pouring into a reservoirs might cause a dam to overflow. This is every power company's nightmare, since in reality means that money are running loose of the mountainside with no one to catch it. Hence they wish to calculate how much water are coming into their reservoirs in the nearby future and also to estimate the level of precipitation at the current, by removing measurement errors. Through improving the short term forecasts of precipitation volumes, the power company may improve their earnings by adjusting the water flow to the power station.

In recent years there has been an extensive build out of weather radars in Norway. Images of the precipitation intensity and wind are available at a time scale as short as 30 minutes. These data are currently not being used to predict the weather, but rather for surveillance of the weather system at the current. Since these images are rather precise compared to satellite images [1], and more importantly shows incredibly well the entire structure of the precipitation field compared to ground based gauges, it is reasonable to think that this can be used for short time precipitation forecasting in meteorology, and equally important as a tool for estimating the amount of water running into a water system. This could prove as an important tool in hydrology and flood forecasting [2], very profitable for both the community and power companies and might result in a reduction in CO_2 emissions since better rainfall estimates improves the production of hydro power. Recently engineers have also been looking at increasing the production of hydro power to compensate when the production from wind mills gets smaller than the demand [3]. To do this, accurate precipitation estimates would be beneficial.

The use of radars enables a lot of data to be analyzed and to be used for predicting the precipitation. There is written some papers on the subject in recent years. This rapport deals with a non-linear state-space filtering, using MCMC. The basic idea is taken from Cornford's article from 2003 [4], which describes the use of basis functions to recreate the precipitation field for a given radar image, and use a simple differential equation for predicting the precipitation at the next time steps. Other papers take on different

approaches, such as [5]. The focus here is on tracking weather storms, but the situation is very similar to ours, using a series of radar images as input. The aim is to track down a weather storm, and further to build a stochastic model that also incorporates birth, death, merging and splitting of targets into the likelihood. Due to the fact that such a model is very complex and that the model is more focused on tracking larger system over vast areas, this approach is not the focus of this paper. However, we will use a simple technique suited for this problem which labels each basis function as active or inactive, where only the active centers are taken forward in time to predict the precipitation. There is also ground based gauges available at various different locations in the middle part of Norway. [2] deals with adjusting the radar image with respect to the gauge data to increase the accuracy of the image by use of block-Kriging. [6] introduces a statistical framework for combining radar reflectivity and gauge measurements to obtain estimates of rainfall rate, taking into account the different sources of error and bias in both sources of data. Here we only do the prediction based on radar images, leaving the adjustment of ground based gauges to further work.

Section 2 gives a presentation of radars and traditional weather forecasting. Here we give a brief introduction of how the data is collected, and which physical relations which govern the atmosphere. In the end of this section, the physics which defines the basis for the model in section 3 is described, and this is summarized as the advection equation. We will assume that the only governing physics of the system is the advection field. Section 3 then defines the Gaussian radial basis functions, which will yield the framework for the precipitation field. Further, a discretization of the advection equation yields a stochastic time series for all basis functions. The full likelihood is presented, before introducing the classification of each basis function as active or inactive. This classification will enable the process of birth and death, yielding a more flexible model. The stochastic time series define a prior to the basis variables, which in section 4 are conditioned on an observed radar image, adjusted and finally taken forward in time. The conditioning step, which is the major time-consuming procedure, is done by modifying the joint MCMC sampler presented in [7]. The first part of the result, section 5, shows the filtering and the prediction result of a computer simulated model. These are used to verify the model, which is tested on a real data set from the Rissa radar located in Sør Trøndelag. The results, shortcomings of the model, and possible further work is summarized in the conclusion given in section 6.

2 Radars and meteorology

This section will give a brief presentation of how a radar works and standard numerical weather forecasting. The last part of the section will present a simple differential equation which later will be used when modeling the precipitation field.

2.1 Radar measuring in a meteorological setting

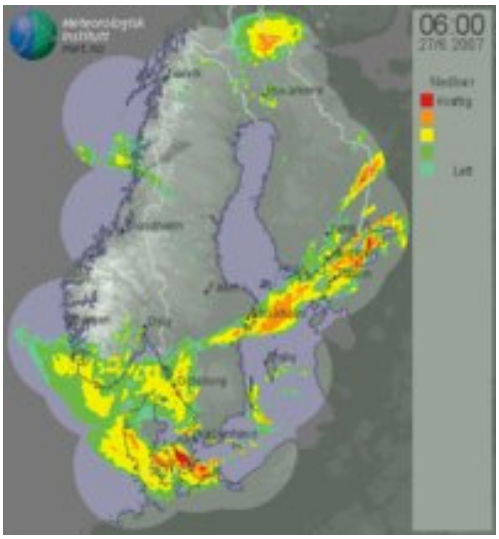
The radar, *radio detection and ranging*, was first used in 1941 [8]. In our setting, radars are used to take measures of wind and precipitation intensity and cover large areas, also where ground and gauge data are not available. Radars used for this purpose are called weather radars. A radar consists of three main parts, transmitter, antenna and receiver. By sending out electromagnetic waves, either radio waves or micro waves, the radar can detect a moving objects motion, direction, altitude and speed.

The radar antenna sends out electromagnetic waves which travel through the air. When these waves hit an object, such as a drop of water falling to the ground, the waves bounce back and are detected by the radar dish. The radar then sends the signals to a computer which in turn calculates the distance, using the time difference from when the signal was emitted to when it was detected. As the distance to an object increases, it gets harder to detect. There are several reasons for this, first is the curvature of the Earth. Secondly, the signal strength decreases proportionally to the distance of and object. If the distance gets to great, the time it takes for a signal to be detected from emission, might be larger than the time to when the next signal is emitted [8]. In later sections, when dealing with radar errors, we will discard these facts and treat all locations with the same amount of error. This is done in order to simplify calculations and develop faster algorithms.

Weather radars send out directed pulses of microwaves which are used to locate precipitation, measuring the precipitation rate R given as millimeter pr. hour, classify it as snow, rain or hail etc. and calculate its motion and height. The motion can be calculated as most new weather radars are pulse-Doppler radars, and as the name implies it uses the Doppler effect to calculate the motion due to the change in frequencies of an object relative to the radar. The grid resolution is typically of distance 1 km by 1 km, yielding a highly accurate spacial orientation. By changing the elevation of the antenna at each rotation, and repeating this process for a number of different angels, the radar is able to scan the volume of the air within its maximum range. This process can take up to 10 minutes and gather data up to 15 km above ground and within a radius of 250 km from the radar. Figure 1(a) shows an image of the radar from which the data in section 5 are collected. It is located in Sør Trøndelag in middle Norway, and is one of eight radars which together cover almost all of Norway's mainland and coastal lines. This can be seen in real time at www.yr.no, which is the joint online weather service from the Norwegian Meteorological Institute and the Norwegian Broadcasting Corporation. Figure 1(b) is a visualization of one such image, were data from Swedish and Danish radars are also

merged.

(a) Rissa radar



(b) www.Yr.no, combined weather images

The calculation of the precipitation intensity itself, also adds to the radar measurement error. When measuring the radar reflectivity(Ze), the emission rate of the microwaves, the weather radar does not measure rainfall rate directly. The rate is calculated through the relation,

$$Ze = \alpha R^{\beta}.$$

In [9], an estimation of the parameters are $\alpha = 200$ and $\beta = 1.6$. However, the conversion equation is derived from the measured drop size distributions which might vary from storm to storm, and even within a storm [9]. We will not address this issue, but the reader is referred to [6], where a method for combining radar reflectivity with gauge data is presented.

In stead of using the usual longitude and latitude degrees to describe a given position on the radar image, there is a widespread use of the UTM coordinates. UTM stands for *Universal Transverse Mercator*, and divides the earth into 60 zones [10]. These zones are determined by the transverse Mercator projection, and use the imaginary line from the north to the south pole, the so called meridian line, as its projection axis instead of the equator. UTM coordinates are easy to handle, and the Pythagorean Theorem can be used as a very good approximation for positioning and distance calculation.

2.2 Numerical weather forecasting

Meteorology is the scientific study of the atmosphere [11]. In recent years, the introduction of powerful computers has enabled the use of numerical calculations in weather forecasting. The movement and behavior of the atmosphere are governed by differential equations from fluid dynamics and thermodynamics. The collection of equations that are used to model the atmospheric flow are called the *primitive equations* [12]. They are a set of nonlinear differential equations which predict the atmospheric flow and are common in most atmospheric modeling [12]. The primitive equations are derived from the fundamental principles given in the next two subsections.

2.2.1 Conservation of momentum and mass, Navier-Stokes equation

Navier-Stokes equation is used in a variety of disciplines within physics. It describes the motion of fluid substances, and can be applied when modeling the atmosphere. In its most general form it can be expressed as,

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \nabla\mathbb{T} + f. \tag{1}$$

Here ρ is the fluids density, $\mathbf{u}$ is the velocity, p is pressure, $\mathbb{T}$ is the average of internal forces, f is a force which acts on the surfaces of the control volume [13]. It can be shown that equation 1 is a statement of the conservation of momentum by applying Newtons second law [13]. Navier-Stokes equation for an incompressible fluid, such as the atmosphere, can be expressed as [14],

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p - \mu\nabla^2\mathbf{u} - \rho g\beta(T - T_0). \tag{2}$$

Here μ is the viscosity, g is the gravitational constant, T is fluid temperature, T_0 is surface temperature and β is a constant proportional to the buoyancy force.

Another statement that often comes with the conservation of momentum is the conservation of mass. In a steady state condition, this statement can be summarized as the rate at which mass enters a system is equal to the rate at which mass leaves the system. Mathematically this is expressed as,

$$\frac{\partial \rho}{\partial t} + \nabla\cdot\rho\mathbf{u} = 0. \tag{3}$$

When the medium is also incompressible, the statement reduces to,

$$\nabla\cdot\mathbf{u} = 0. \tag{4}$$

Equation 3 is a special case of the continuity equation for a fluid in steady state motion. This equation will found the basis for the model described in section 3. Equations 3 and 2 are often referred to as the Navier-Stokes equations and are together with Thermal energy equation essential in modeling the atmosphere.

2.2.2 Conservation of thermal energy, the thermal energy equation

Together with conservation of momentum and mass, the thermal energy also has to be conserved within the system. Heat transfer always take place from a region with higher temperature to a region with lower temperature. A property called the thermal conductivity, k, relates the rate of thermal energy per unit area, $\mathbf{q}$, to the temperature gradient, ∇T. Through physical experiments a proportional relation between these units has been verified, and it is known as *Fourier's law of heat conduction* [13],

$$\mathbf{q} = -k\nabla T. \tag{5}$$

The first law of thermodynamics states that the change of internal thermal energy in the system is equal to the heat added to the system, plus the work done by the system. This can be mathematically expressed as,

$$dU = \delta Q - \delta W. \tag{6}$$

Here dU is a small increase in the internal energy of the system, δQ is a small amount of heat added to the system, and δW is a small amount of work performed by the system [13]. It is shown in [14], that using the first law of thermodynamics, one can obtain the following partial differential equation for an incompressible fluid,

$$\rho c_p \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot T \right) = \nabla \cdot k\nabla T + q. \tag{7}$$

Here c_p is the specified heat capacity for the fluid, and q indicates the internal energy for a small element of the fluid.

We will not go further into numerical weather forecasting here than to say that the equations 2,3 and 7 found the basic when modeling, solving and predicting the state of the atmosphere. For further reading regarding numerical weather prediction see [15]. In the next subsection we will instead look at a series of radar images, a situation were data for pressure and temperature is not available but instead the precipitation intensity, and use stochastic variables to obtain a simple model of the precipitation field at a given image of the atmosphere.

2.3 The advection equation

Let us now assume that we have a time series of radar images. An example of a computer generated sequence of radar images according to the framework that will be presented in section 3 can be seen from figure 1. The sequence of images comes in frequently, and the aim is to describe the precipitation field in a 0-3 hour ratio. Due to the frequent image series, it is possible to make a number of simplifications/linearizations of the physics governing precipitation field. One important approximation is to say that there is no imminent loss or gain of precipitation between two observations of the time series. This will in most situations not be a correct assumption, but for now it work as satisfying

approximation, and section 3 will later deal with the issue of a net gain or loss of precipitation. All external forces are left out, since the data available does not contain any information but the rate of precipitation.

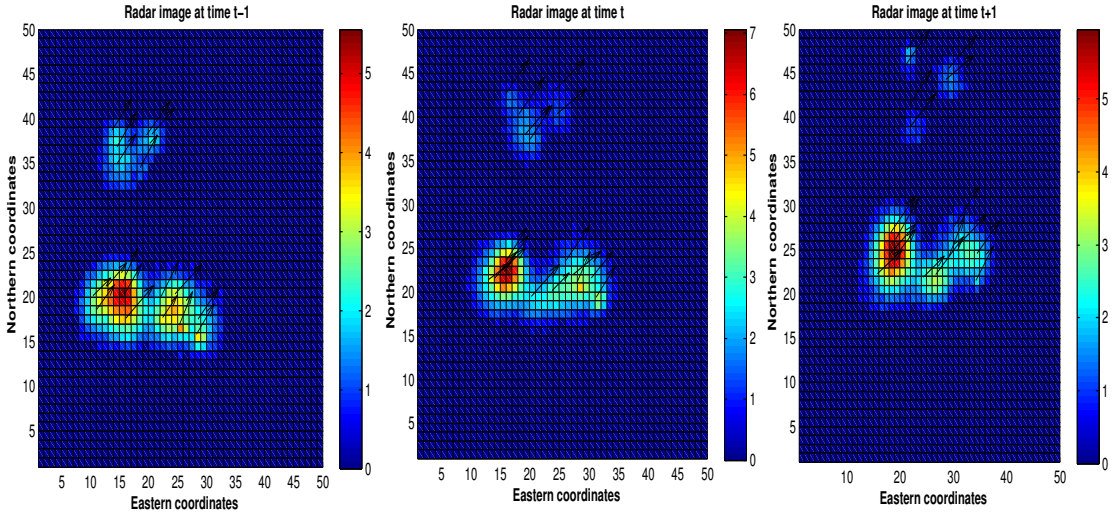


Figure 1: An image of computer generated radar images for three time steps. The advection is shown with arrows. The lighter colors indicate locations with high intensities of precipitation, given as millimeter pr. hour.

It is obvious however, that the air carrying the precipitation field, $\mathbf{R}$, might be moving in one direction and the precipitation is drifting(diffusing) in different directions. Such transport, where certain properties are conserved within a medium, in our case the total amount of precipitation, is called *advection*. In other words, the advection field, $\mathbf{u}$ describes the net flow of the medium [13], and is given as a vector field. We will assume that there might be some variation across the advection field, hence the transport is not in a constant direction.

Conservation of the total amount of precipitation equals the conservation of mass. This information is summarized in equation 8, and deals with the movement of the precipitation given the advection field,

$$\frac{\partial \mathbf{R}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{R} \approx 0. \tag{8}$$

In words this equation tells us that, the rate at which the precipitation changes with respect to time, equals the amount of precipitation moving in or out of a given control area.

3 Model

3.1 Hidden Markov models

As in the previous section, and in many other occasions, mathematicians have modeled temporal phenomena in nature or through man made activities, using differential equations. We however will take on a statistical approach, trying to combine the use of time series with the use of a partial equation. Hidden Markov models have proven to be a successful approach of combining such models [16]. By introducing unknown(hidden) stochastic parameters that are only dependent on the previous state(Markov), we will try to recreate the precipitation field at the current time, given a series of previous radar images.

In the general case, let x_t denote the unknown state for any given system. The data is given as d_t. Assume Markovian property for the prior for the state model. The dependency structure of a 2-level hidden Markov Model is shown in dag in figure 2.

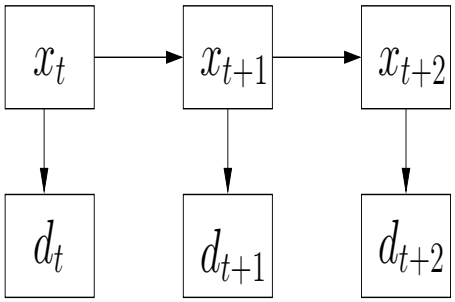


Figure 2: A hidden Markov model illustrated as a dag. The hidden state is x_t at time t, and d_t is the corresponding observation. Data are conditionally independent, given the states as we have no arrows between the d_t values. Hidden states are dependent by the dynamical model, indicated by the arrows from x_t to x_{t+1}.

A formal model with additive Gaussian noise can now be expressed as,

$$\begin{aligned} x_t &= f(x_{t-1}) + \epsilon_x \\ d_t &= g(x_t) + \epsilon_d, \end{aligned} \tag{9}$$

where the error terms are multinormal distributed,

$$\begin{aligned} \epsilon_x &\sim N(0, \Sigma_x) \\ \epsilon_d &\sim N(0, \Sigma_d). \end{aligned}$$

We will get back to the sequential conditioning and forecasting for the state of a given time series of data, in section 4. This section specify the full model, obtaining the priors for the state model and the likelihood equation for one radar image given the state.

3.2 Basis function for precipitation field

Let us now go back to the situation were we have a precipitation field measured by a radar. Label the observed radar image as $\mathbf{I}_t$, and let $\mathbf{R}_t$ be the true underlying precipitation field, both being measured in mm. pr. hour. Hence $\mathbf{I}_t$ will yield the data, and $\mathbf{R}_t$ will yield the state of the system according to the framework presented in section 3.1.

We must first find a representation of $\mathbf{R}_t$ which yield the hidden state model according to the framework presented in section 3.1. In its simplest form, the representation of the model framework should be able to recreate $\mathbf{R}_t$. The idea presented in [4] is to represent the precipitation field as a sum of N, of basis functions. Each basis function is constructed as a Gaussian radial function with a corresponding two dimensional center, radius and intensity. The entire precipitation field is obtained by taking the sum of such basis functions. At a given time step t, we define the variables,

$$
\begin{aligned}
\mathbf{x} &= [x, y] \text{ , position vector,} \\
\mathbf{c}_t &= [\mathbf{c}_{t_x}, \mathbf{c}_{t_y}] = \begin{bmatrix} c_{t_{x_1}} & c_{t_{y_1}} \\ \vdots & \vdots \\ c_{t_{x_N}} & c_{t_{y_N}} \end{bmatrix} \text{ , center coordinates,} \\
\mathbf{h}_t &= [h_{t_1}, \cdots, h_{t_N}]' \text{ , intensities,} \\
\mathbf{w}_t &= [w_{t_1}, \cdots, w_{t_N}]' \text{ , radii.}
\end{aligned}
$$

The precipitation at time t, given position $\mathbf{x}$, can then be calculated as,

$$
\mathbf{R}_t(\mathbf{x}) = \sum_{n=1}^{N} h_{t_n} exp\left[-\frac{1}{2} \frac{(x - c_{t_{x_n}})^2 + (y - c_{t_{y_n}})^2}{w_{t_n}^2} \right]. \tag{10}
$$

An illustration of a few basis functions are given in figure 3. By choosing a sufficient number of basis functions, any real precipitation field could potentially be approximately recreated in terms of equation 10.

Since both all elements $\mathbf{w}_t$ and $\mathbf{h}_t$ must clearly be larger than zero, the representation of $\mathbf{R}_t(\mathbf{x})$ in terms of equation 10 means that it is never really zero. This is off course not an accurate assumption, and the precipitation level should therefore be set to zero below a given limit, $\mathbf{R_{crit}}$. By choosing ellipses instead of circles when constructing basis functions, one should expect to get a more flexible representation of the actual precipitation field. However, for computational reasons, we choose to use circles and and assume that this will give a sufficient description of the field.

3.3 Model dynamics, discretization and prior distributions

In a short term setting, the dynamics is simplified such that no precipitation is set to be generated, and no precipitation is set to disappear. This is off course not the case in

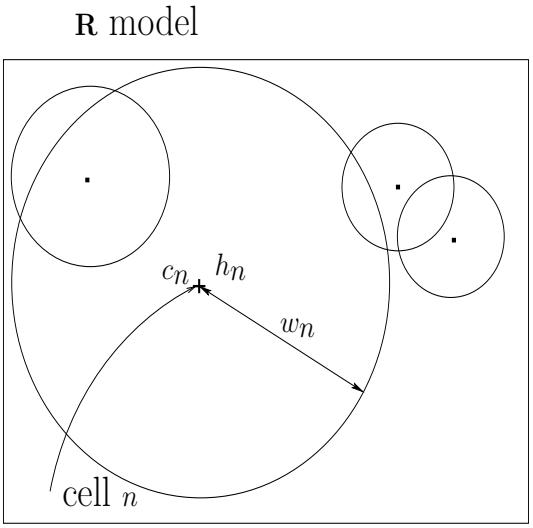


Figure 3: Representation of precipitation field $\mathbf{R}_t$, given by radial basis functions. In the representation c_n is the center, w_n is the radius and h_n is the intensity for a specified basis function n.

almost any true scenario, but will be fairly accurate in a short time frame. The system equation is as before given by the advection equation 8, from section 2.3,

$$\frac{\partial \mathbf{R}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{R} \approx 0. \tag{11}$$

We will now look at how equation 11 acts on the parameters c_{t_n}, w_{t_n} and h_{t_n}, for $n = 1, \ldots, N$. After adding random noise to each parameter, the resulting discretization will yield the prior distributions to state of the model according to Bayesian terminology.

3.3.1 Prior distribution for basis radii, $\mathbf{w}_t$

The situation for the radii w_{t_n} for $n = 1, \ldots, N$, is simple. On the basis of the physics described in the advection equation 11, we do not expect any of the w_{t_n} to change from one time step to another since we do not expect any growth or decay of the precipitation. Adding an extra error term allows for a small change in the extent of the basis functions between each time step. This allows for a slight growth or decay of the precipitation field as a function of time,

$$\begin{aligned} \mathbf{w}_{t+1} &= \mathbf{w}_t + \boldsymbol{\epsilon}_{\mathbf{w}} \\ \boldsymbol{\epsilon}_{\mathbf{w}} &\sim N\left(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}}\right). \end{aligned} \tag{12}$$

Here $\boldsymbol{\Sigma}_{\mathbf{w}}$ is a properly chosen diagonal covariance matrix, having σ_w^2 as nonzero elements. The dependence structure is simple and is shown by figure 4.

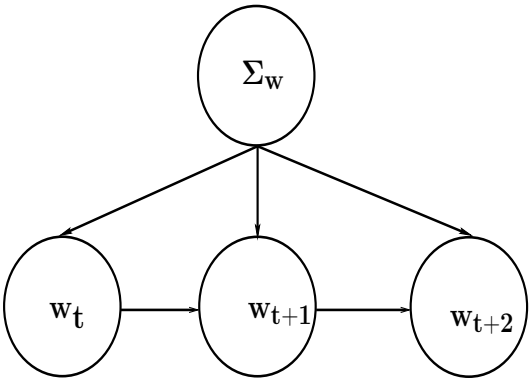


Figure 4: Dependency structure for the radii $\mathbf{w}_t$ of the basis functions. The next state is only dependent on the current state and a common constant covariance matrix $\mathbf{\Sigma_w}$.

3.3.2 Prior distribution for basis intensities, $\mathbf{h}_t$

The situation for the intensities h_{t_n} for $n = 1, \ldots, N$ is similar. On the basis of the physics described by equation 11, none of the h_{t_n} is expected to change from one time step to another. But to allow for some gain of strength or decay in the intensity we add some random noise. Hence we get,

$$\begin{aligned} \mathbf{h}_{t+1} &= \mathbf{h}_t + \boldsymbol{\epsilon}_\mathbf{h} \\ \boldsymbol{\epsilon}_\mathbf{h} &\sim N(\mathbf{0}, \mathbf{\Sigma_h}). \end{aligned} \tag{13}$$

The dependence structure of the intensities is shown in figure 5.

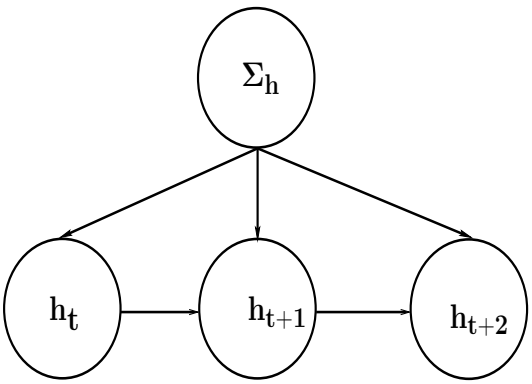


Figure 5: Dependency structure for the intensities $\mathbf{h}_t$ of the basis functions. The next state is only dependent on the current state and a common constant covariance matrix $\mathbf{\Sigma_h}$.

3.3.3 Prior distributions for basis centers, $\mathbf{c}_t$, and advection, $\mathbf{u}_t$

It is clear from equation 11 that the movement of the centers, $\mathbf{c}_t$, are solely decided by the advection field $\mathbf{u}_t$ which acts as the velocity field of the basis functions and is assumed to be locally constant. Since this linearization is not entirely correct, we add some random noise when performing the discretization of the position vector of the center at each time step. The change in the northern y, direction and the eastern x, direction are assumed independent. The discretization then becomes,

$$\begin{aligned} \{\mathbf{c}_{t+1}\}_{\mathbf{x},\mathbf{y}} &= \{\mathbf{c}_t + \delta_t \cdot \mathbf{u}_{t+1} + \boldsymbol{\epsilon}_{\mathbf{c}}\}_{\mathbf{x},\mathbf{y}} \\ \boldsymbol{\epsilon}_{\mathbf{c}} &\sim N(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{c}}). \end{aligned} \tag{14}$$

Here δ_t is the length of the time step. The covariance matrix $\boldsymbol{\Sigma}_{\mathbf{c}}$, is diagonal where all nonzero elements are of size σ_c.

The advection field $\mathbf{u}_{t+1}$ is assumed to be locally constant at each time step when updating the basis functions. After this update, one can allow for a change in the advection in the same manner as before by introducing an error term $\boldsymbol{\epsilon}_{\mathbf{u}}$, holding the basis functions constant. Assuming that the change in the eastern direction, x, and northern direction, y is independent, this yield,

$$\begin{aligned} \{\mathbf{u}_{t+1}\}_{\mathbf{x},\mathbf{y}} &= \{\mathbf{u}_t + \boldsymbol{\epsilon}_{\mathbf{u}}\}_{\mathbf{x},\mathbf{y}} \\ \boldsymbol{\epsilon}_{\mathbf{u}} &\sim N(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{u}_t}). \end{aligned} \tag{15}$$

It is important that $\boldsymbol{\Sigma}_{\mathbf{u}_t}$ is a full covariance matrix with positive elements. This is clear since the atmosphere tends to drift in the same direction over large areas, and if there is a rapid change at one time step, this change is assumed to be somewhat uniform over the entire grid. The advection at one geographical coordinate must be fairly equal to the advection at a nearby location. Therefore the correlation structure for the advection vectors $\mathbf{u}_t$ is chosen so that it drops of with an exponential rate proportional to the distance to the next center. The covariance matrix $\boldsymbol{\Sigma}_{\mathbf{u}_t}$, is then a function of the centers,

$$\boldsymbol{\Sigma}_{\mathbf{u}_t} = \sigma_u^2 \cdot \exp\begin{bmatrix} 0 & (-\frac{d_{t_{1,2}}}{a}) & \cdots & (-\frac{d_{t_{1,N-1}}}{a}) & (-\frac{d_{t_{1,N}}}{a}) \\ (-\frac{d_{t_{2,1}}}{a}) & 0 & \cdots & (-\frac{d_{t_{2,N-1}}}{a}) & (-\frac{d_{t_{2,N}}}{a}) \\ \vdots & & \ddots & & \vdots \\ (-\frac{d_{t_{N-1,1}}}{a}) & (-\frac{d_{t_{N-1,2}}}{a}) & \cdots & 0 & (-\frac{d_{t_{N-1,N}}}{a}) \\ (-\frac{d_{t_{N,1}}}{a}) & (-\frac{d_{t_{N,2}}}{a}) & \cdots & (-\frac{d_{t_{N,N-1}}}{a}) & 0 \end{bmatrix}. \tag{16}$$

The drop off in correlation is given as a. Each $d_{t_{i,j}}$ is the Euclidean distance between center i to center j at time t, and $\boldsymbol{\Sigma}_{\boldsymbol{u}_t}$ then secures that centers close to each other move in a similar direction. This leads to a dependency from the centers to the advection field. This is shown in figure 6, giving the dependence structure between $\mathbf{c}_t$ and $\mathbf{u}_t$.

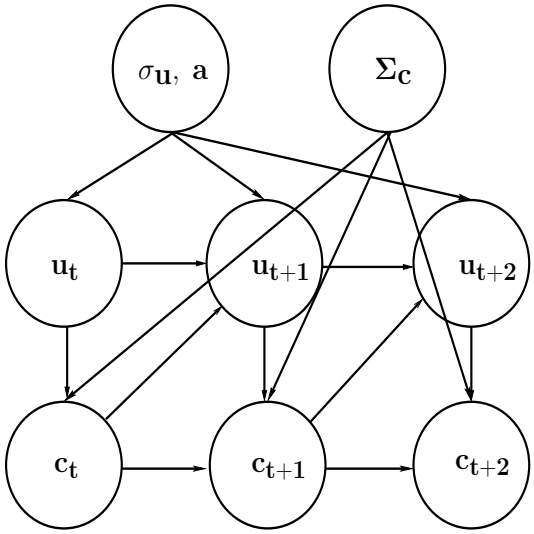


Figure 6: Dependency structure between centers $\mathbf{c}_t$ and advection $\mathbf{u}_t$. The next state of the advection, $\mathbf{u}_{t+1}$, is dependent on the current state, $\mathbf{u}_t$, and its covariance matrix, $\mathbf{\Sigma_u}$, which is calculated at each time step, t, by correlation parameter a, the standard deviation σ_u, and the distance between centers $\mathbf{c}_t$. The next state for the centers, $\mathbf{c}_{t+1}$, is dependent on the current state, $\mathbf{c}_t$ and the constant covariance matrix, $\mathbf{\Sigma_c}$, in addition to the advection, $\mathbf{u}_{t+1}$. All dependencies are indicated with arrows.

After the prior distributions have been specified in section 3.3, where the dependence structure between the basis functions is visualized in figures 4, 5 and 6. We can now build the complete hierarchical model. Given the specification of $\mathbf{R}_t$ in terms of $\mathbf{c}_t$, $\mathbf{w}_t$ and $\mathbf{h}_t$, the update via the advection $\mathbf{u}_t$ is shown in figure 7.

3.4 Likelihood

The radar image contains M number of discretization in each direction, together making up a $M \times M$ sized 2-dimensional grid $\mathbf{X}$, where each pixel contains a radar measurement at a given (x, y) coordinate. The radar grid $\mathbf{X}$ consists of equally spaced pixels, on this $M \times M$ sized radar grid. For computational reasons $\mathbf{X}$ is given as a one dimensional vector with length M^2,

$$\mathbf{X} = [(x_1, y_1), (x_1, y_2), \cdots, (x_1, y_M), (x_2, y_1), \ldots, (x_2, y_M), (x_3, y_1) \ldots, (x_M, y_M)]'. \quad (17)$$

All the measurements are from now on assumed normal, independent and identically distributed (i.i.d.),

$$\mathbf{I}_t(\mathbf{X}) = \mathbf{R}_t(\mathbf{X}) + \boldsymbol{\epsilon}_I$$
$$\boldsymbol{\epsilon}_I \sim N(\mathbf{0}, \mathbf{\Sigma}_I) \quad (18)$$
$$\mathbf{\Sigma}_I = diag(\sigma_I^2). \quad (19)$$

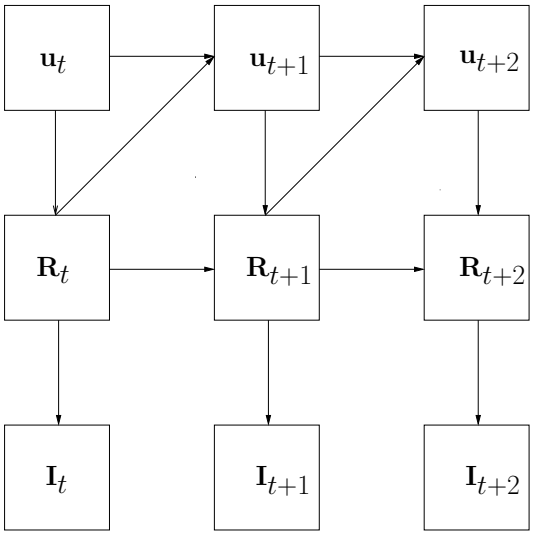


Figure 7: The hierarchical structure of the stochastic parameters specified in the model, all constants are left out. The next state of $\mathbf{u}_{t+1}$ is dependent of the current state $\mathbf{u}_t$ and the state of the precipitation field, $\mathbf{R}_t$. The next state of the precipitation field ,$\mathbf{R}_{t+1}$, is dependent of the current state, $\mathbf{R}_t$, and the next update of $\mathbf{u}_{t+1}$. All dependency relations are indicated with arrows.

Here the diagonal $M \times M$ covariance matrix $\mathbf{\Sigma}_I$, is independent of time and spatial orientation. This is not a true assumption since the error in the radar image is highly dependent of the intensity of the precipitation at a given time, and geographical obstacles and distortions, as explained in section 2.1. For simplicity this has been discarded, but for future work this should be implemented in a full model. For example, [1] suggest to estimate the covariance matrix $\mathbf{\Sigma}_I$ by comparison with ground based gauge data.

We will now look at the likelihood function $L(\mathbf{I}_t|\mathbf{c}_t, \mathbf{w}_t, \mathbf{h}_t)$ at one given time step, for the grid defined by equation 17. We assume spatial independence, that all of the errors in the radar measurements, $\boldsymbol{\epsilon}_I$, are independent from the others. Hence we get,

$$L(\mathbf{I}_t|\mathbf{c}_t, \mathbf{w}_t, \mathbf{h}_t) \propto \prod_{i,j=1}^{M} exp\left\{ -\frac{\left(\mathbf{I}(x_i, y_j) - \sum_{n=1}^{N} h_{t_n} exp\left[-\frac{1}{2}\frac{(x_i - c_{t_{n_x}})^2 + (y_j - c_{t_{n_y}})^2}{w_{t_n}^2}\right]\right)^2}{2 \cdot \sigma_I^2} \right\}.$$

This is a rather expensive function to calculate. It is therefore useful to write the likelihood as a product of vectors. Define the Kronecker vector of the x and y coordinates,

$$\mathbf{X}_{Kron} = [x_1, \ldots, x_M, x_1, \ldots, x_M, \ldots, \ldots, x_1, \ldots x_M]', \quad M^2 \times 1 \text{ vector}, \tag{20}$$

$$\mathbf{Y}_{Kron} = [y_1, \ldots, y_1, y_2, \ldots, y_2, \ldots, \ldots, y_M, \ldots, y_M]', \quad M^2 \times 1 \text{ vector}, \tag{21}$$

and convert the measured radar image $M \times M$ matrix $\mathbf{I}_t$, to a $M^2 \times 1$ vector,

$$\hat{\mathbf{I}}_t = \mathbf{I}_t(x_1, y_1), \mathbf{I}_t(x_1, y_2), ..., \mathbf{I}_t(x_1, y_M), \mathbf{I}_t(x_2, y_M), ...\mathbf{I}_t(x_M, Y_M). \tag{22}$$

Let $\mathbf{O}_{M^2 \times 1}$ be a $M^2 \times 1$ vector of ones. The logarithm of the likelihood can now be written as,

$$logL(\mathbf{I}_t|\mathbf{c}_t, \mathbf{w}_t, \mathbf{h}_t) = -\frac{1}{2}(\hat{\mathbf{I}}_t - \sum_{n=1}^{N} \mathbf{R_n})' {\mathbf{\Sigma_I}}^{-1} (\hat{\mathbf{I}}_t - \sum_{n=1}^{N} \mathbf{R_n}). \tag{23}$$

Here $\mathbf{R_n}$ is a $M^2 \times 1$ vector defined so that,

$$\mathbf{R}_n = h_n \cdot \mathbf{O}'_{M^2 \times 1} exp\left[-\frac{1}{2}(\mathbf{X}_{cron} - c_{n_x} \cdot \mathbf{O}_{M^2 \times 1}).^2 + (\mathbf{Y}_{cron} - c_{n_y} \cdot \mathbf{O}_{M^2 \times 1}).^2\right]. \tag{24}$$

The dot in the squared expressions in equation 24 denotes that the elements of the vectors should be squared, not the vector itself. The likelihood function is still rather expensive to calculate, but is now written as a sum of vector products which drastically reduces the time it takes to compute it.

3.5 Priors at time t_0

Finally, to have a complete model we need to decide the prior distributions at initial time t_0. We have no prior information about the basis functions of the precipitation field, except that radii and intensities must be larger than zero. We start by defining uniform prior distributions for the basis variables on legal intervals,

$$\begin{array}{rcll} w_{t_{0n}} \sim & U[0, \infty] & n = 1, ..., N \\ h_{t_{0n}} \sim & U[0, \infty] & n = 1, ..., N \\ c_{t_{0n_x}} \sim & U[x_1, x_m] & n = 1, ..., N \\ c_{t_{0n_y}} \sim & U[y_1, y_m] & n = 1, ..., N. \end{array} \tag{25}$$

One could argue that the priors for $\mathbf{w}_0$ and $\mathbf{h}_0$ could be more restricted and therefore given a more informative distribution such as a gamma or log normal distribution. Also one could expect the position of the centers to be correlated, but for computational reasons and the rather extensive amount of information in the likelihood, this is discarded.

The only prior knowledge we have about the advection is that the magnitude and direction has a strong correlation. Therefore we choose,

$$\mathbf{u}_{t_1} \sim N(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{u}_{t_0}}). \tag{26}$$

This means that the prior at the initial state, for the advection field this is t_1, is basically the same as for the other time step from equation 26. The only difference is that the mean is set to zero since we have no previous value to compare with.

3.6 Model modifications, inactive basis functions,birth and death

The model, as it is described in previous sections, is somewhat incomplete. We can not have any new rain showers entering the system, have any birth or death of rain showers but all variables are bound to the initial image at t_0. There is also another issue that must be addressed. That is, what if a basis function n, has too low intensity h_n to overcome the critical lower value $\mathbf{R_{crit}}$, or too narrow radius to be calculated at a grid location. This means that the current basis function is hidden, or as we will call it, *inactive*.

Remember that the precipitation level is only calculated at fixed grid locations. This might cause problems if the radius of a basis function gets to small. As seen from figure 8, a basis functions might not contribute to any of the nearby pixels, even though its corresponding intensity could be of a large value. This might cause problems at the next time step, when this large value might be calculated at a grid location. The problem

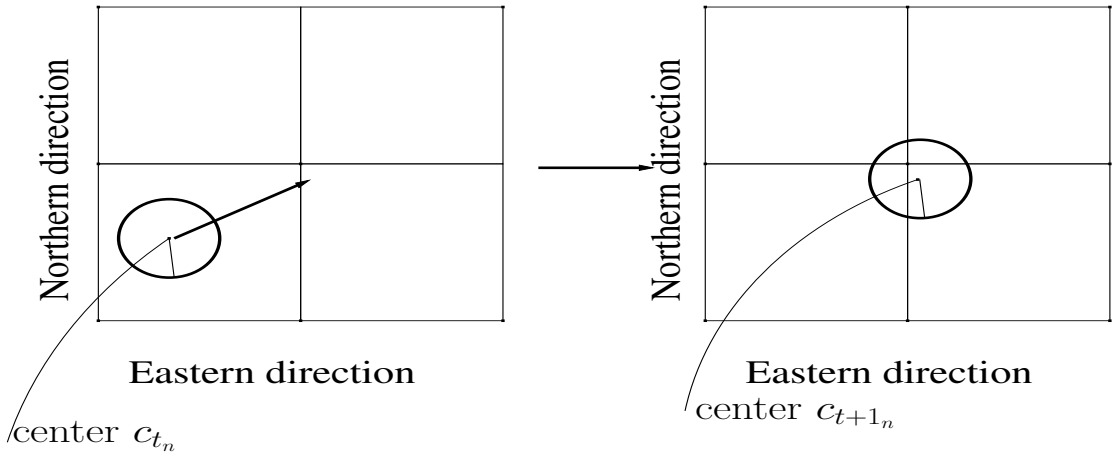


Figure 8: To the left we see a specified basis function n, with a high intensity but small radius, getting caught in between two neighboring pixels. When updating to next location c_{t+1_n}, the figure to the right shows that the contribution from basis function n will greatly differ even though the specified basis function is basically of the same magnitude in both situations.

with a given radius w_{t_n}, getting to small is solved simply by setting a proper lower limit for any w_{t_n},

$$w_{t_n} > \mathbf{w_{crit}}, \text{ for } t = 0, \ldots, T \text{ and } n = 1, \ldots, N. \tag{27}$$

To secure better control with the location of the basis center c_{t_n}, they are strictly defined to be within the grid,

$$c_{t_j} \in \mathbf{X}, \text{ for } t = 0, \ldots, T \text{ and } n = 1, \ldots, N. \tag{28}$$

Assume further that the requirements of equations 27 and 28 are fulfilled. For a given basis function n at time t, imagine that the precipitation level at the position of the corresponding center $\mathbf{R}_t(c_{t_n})$ is below the critical value,

$$\mathbf{R}_t(c_{t_n}) < \mathbf{R}_{\mathbf{crit}}. \tag{29}$$

In fact this means that the contribution from basis function n, to $\mathbf{R}_t$, is of no importance at all. The basis function is then defined to be inactive. Further define the vector,

$$\mathbf{A}_t = [A_{t_1}, \ldots, A_{t_n}, \ldots, A_{t_N}],$$

containing binary variables, 0 indicating an inactive variable. Since an inactive basis function do not contribute to $\mathbf{R}_t$, it is completely redundant, and it contains no information of which should be taken into account into the next update. If for a given time t and for a specified basis function n, $A_{t_n} = 0$, this equals the situation at t_0, and the prior values for the basis functions at the next time step are therefore reset to the value specified in section 3.5,

$$\left.\begin{array}{rcl} w_{t+1_n} & \sim & U[0,\infty] \\ h_{t+1_n} & \sim & U[0,\infty] \\ c_{t+1_{n_x}} & \sim & U[x_1, x_m] \\ c_{t+1_{n_y}} & \sim & U[y_1, y_m] \end{array}\right\} \text{if } A_{t_n} = 0. \tag{30}$$

Asserting a basis function as inactive, corresponds to resetting the basis functions to the initial conditions at the next time step. This could mean that a rain shower has died out, and the basis function can then be moved freely around the image, possibly resulting in a new rain shower somewhere else on $\mathbf{X}$. Modifying the model by introducing the active variable $\mathbf{A}_t$, has then allowed for birth and death of new rain showers, not just some slight growth or decay of the initialized image.

For the advection at the next time step $\mathbf{u}_{t+1}$, the situation is a bit more complex. If a basis function is declared as inactive, it means that it does not contribute to the image $\mathbf{R}_t$ and has no corresponding u_{t+1_j}. In other words the advection can not be updated at the current center at that time. The stochastic binary vectors $\mathbf{A}_0, \ldots, \mathbf{A}_T$ might vary, and the number of advection vectors equals the sum of each of $\mathbf{A}_t$ at that specified time step. Hence we have a different dimension parameter space for each $\mathbf{u}_t$. So there is now two possible situations for each u_{t+1_n} given previous $\mathbf{A}_t$ and $\mathbf{u}_{t+1}$,

$$u_{t+1_n} = \begin{cases} u_{t_n} + \epsilon_u & \text{if } A_{t_n} = 1 \\ \\ 0 & \text{if } A_{t_n} = 0. \end{cases} \tag{31}$$

If $A_{t_n} = 1$ and $A_{t-1_n} = 0$, the previous u_{t_n} is not yet specified. Since the advection are highly correlated, we choose to set this value to the mean of the advection of the active functions,

$$u_{t_n} = \frac{1}{N_a} \sum_{i=1}^{N_a} u_{t_i} \text{ , if } A_{t_n} = 1 \text{ and } A_{t-1_n} = 0, \tag{32}$$

were N_a is the number of active basis functions at time t.

The covariance matrix $\mathbf{\Sigma}_{u_t}$ is almost as before,

$$\mathbf{\Sigma}_{u_t} = \sigma_u^2 \cdot \exp \begin{bmatrix} 0 & (-\frac{d_{1,2}}{a}) & \cdots & (-\frac{d_{1,N_a-1}}{a}) & (-\frac{d_{1,N_a}}{a}) \\ (-\frac{d_{2,1}}{a}) & 0 & \cdots & (-\frac{d_{2,N_a-1}}{a}) & (-\frac{d_{2,N_a}}{a}) \\ \vdots & & \ddots & & \vdots \\ (-\frac{d_{N_a-1,1}}{a}) & (-\frac{d_{N_a-1,2}}{a}) & \cdots & 0 & (-\frac{d_{N_a-1,N_a}}{a}) \\ (-\frac{d_{N_a,1}}{a}) & (-\frac{d_{N_a,2}}{a}) & \cdots & (-\frac{d_{N_a,N_a-1}}{a}) & 0 \end{bmatrix}, \quad (33)$$

the only difference being N being replaced by N_a. The model has now become more flexible, but it does come with a prize. As we shall see in the next section, this goes on the expense of the traditional filtering methods, which is often used to condition and forecast a hidden Markov chain.

4 Filtering techniques and posterior distributions

The aim is to obtain the forecast and the conditioning of the precipitation field $\mathbf{R}(\mathbf{c}_t, \mathbf{w}_t, \mathbf{h}_t)$, given radar data arriving with time. Express the basis variables as,

$$\begin{aligned}\left[\mathbf{c}^u_{t+1}, \mathbf{w}^u_{t+1}, \mathbf{h}^u_{t+1}\right] &= \left[\mathbf{c}_{t+1}, \mathbf{w}_{t+1}, \mathbf{h}_{t+1} | \mathbf{I}_{0:t}\right] \\ \left[\mathbf{c}^c_{t+1}, \mathbf{w}^c_{t+1}, \mathbf{h}^c_{t+1}\right] &= \left[\mathbf{c}_{t+1}, \mathbf{w}_{t+1}, \mathbf{h}_{t+1} | \mathbf{I}_{0:t+1}\right].\end{aligned}$$

The hyper index u indicates the forecast, and stands for *unconditioned*. The hyper index c indicates the filtering step, and stands for *conditioned*. For notational purposes later in this section, let

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{c}_t \\ \mathbf{w}_t \\ \mathbf{h}_t \end{bmatrix}. \tag{34}$$

From the framework presented in the previous section, we have also introduced the advection field which needs to be analyzed in the same manor,

$$\begin{aligned}\mathbf{u}^u_{t+1} &= \left[\mathbf{u}_{t+1} | \mathbf{I}_{0:t}\right] \\ \mathbf{u}^c_{t+1} &= \left[\mathbf{u}_{t+1} | \mathbf{I}_{0:t+1}\right].\end{aligned}$$

In state space modeling there exists various filtering techniques for obtaining this forecast and conditioning. The Markovian property enables us to update the forecast and the conditioning recursively, which is very convenient when data is arriving online. This section first presents some basic filters, and finally a MCMC scheme for filtering within the Bayesian framework.

4.1 Traditional Kalman Filter

Consider a dynamic system, modeled as a hidden Markov model via the linear state space model with additive Gaussian noise,

$$\begin{aligned}x_t &= G_t x_{t-1} + \epsilon_x \\ d_t &= H_t x_t + \epsilon_d,\end{aligned} \tag{35}$$

G_t and H_t being linear operators(matrices). The conditioned and unconditioned variable are then also Gaussian,

$$\begin{aligned}x^u_t &= [x_t | d_0, ..., d_{t-1}] \sim N(\mu^u_t, \Sigma^u_t), \\ x^c_t &= [x_t | d_0, ..., d_t] \sim N(\mu^c_t, \Sigma^c_t).\end{aligned}$$

The normality is a result of the linear model given in equation 35. The exact solution for the mean and the variance is obtained from the Kalman filter algorithm. As described in [16], it is given as algorithm 1. As one can see from algorithm 1, the expected values and uncertainties can be analytically calculated recursively using the Gauss linear relations. Unfortunately our model can not be resolved this neatly, but the idea of using the previous forecast to update the conditioning suits the situation very well.

Algorithm 1 Traditional Kalman Filter

Initiate:
$\boldsymbol{\mu}_0^u = \boldsymbol{\mu}_0^x$
$\boldsymbol{\Sigma}_0^u = \boldsymbol{\Sigma}_0^x$
for t=1 to T **do**
Conditioning:
$\boldsymbol{\mu}_t^c = \boldsymbol{\mu}_t^u + \boldsymbol{\Sigma}_t^u \mathbf{H}_t' \left[\mathbf{H}_t' \boldsymbol{\Sigma}_t^u \mathbf{H}_t' + \boldsymbol{\Sigma}_t^d\right]^{-1} (\mathbf{d}_t - \mathbf{H}_t \boldsymbol{\mu}_t^u)$
$\boldsymbol{\Sigma}_t^c = \boldsymbol{\Sigma}_t^u - \boldsymbol{\Sigma}_t^u \mathbf{H}_t' [\mathbf{H}_t \boldsymbol{\Sigma}_t^u \mathbf{H}_t' + \boldsymbol{\Sigma}_t^d]^{-1} \mathbf{H}_t \boldsymbol{\Sigma}_t^u$
Forecasting:
$\boldsymbol{\mu}_{t+1}^u = \mathbf{A}_t \mu_t^c$
$\boldsymbol{\Sigma}_{t+1}^u = \mathbf{A}_t \boldsymbol{\Sigma}_t^c \mathbf{A}_t' + \boldsymbol{\Sigma}_t^x$
end for

4.2 Kalman related filters

4.2.1 Extended Kalman filter (EKF)

In many cases there are deviations from the Gauss linearity in the model. When such deviations occur, either in the state equation or in the likelihood, a simple approach is the extended Kalman filter. The problem with the nonlinearity is then solved by taking the Taylor expansion of first order, converting the system to a Gauss linear one. This corresponds to linearizing the current mean and covariance. In many cases however, this is not sufficient for obtaining a good approximation, making the extended Kalman filter a poor solution to the problem [16]. This is also the case for us, since the high dimensional precipitation function $\mathbf{R}_t(\mathbf{x})$ given in equation 10, is not suited for linearization.

4.2.2 Ensemble Kalman filter (EnKF)

In the case of a nonlinear state equation, but where the error term/likelihood for the data is linear, the ensemble Kalman filter has in many situations proven to be a very efficient filter. The basic idea is to generate *ensembles* from the state equation, adjusting it according to the data model. The update is done by estimating the covariance structure between the ensembles and the data, assuming Gaussian distributions. Consider a nonlinear state equation, x_t, and a Gauss linear data model, d_t,

$$x_t = f(x_t|x_{t-1}) + \epsilon_x \ \text{, where } \epsilon_x \sim N(0, \Sigma_x), \tag{36}$$
$$d_t = H_t x_t + \epsilon_d \ \text{, where } \epsilon_d \sim N(0, \Sigma_d). \tag{37}$$

Algorithm 2, the ensemble Kalman filter, is also taken from [16]. The ensemble Kalman filter replaces the covariance matrix in the traditional Kalman filter with the sample covariance, under the assumption that all probability distributions involved are Gaussian. When this approximation is sufficient, the EnKF performs very well and has a satisfying runtime. But due to the nonlinear nature of the likelihood from equation 20,

Algorithm 2 Ensemble Kalman Filter

Initiate:
N_e =no. of ensembles
$\mathbf{x}_0^{u(i)};\ i = 1, \ldots, N_e$
$\boldsymbol{\epsilon}_0^{d(i)} \sim N_{p_d}(\mathbf{0}, \mathbf{I}_{p_d});\ i = 1, \ldots, N_e$
$\mathbf{d}_0^{(i)} = f(x_0^{u(i)}, \epsilon_0^{d(i)});\ i = 1, \ldots, N_e$
$\mathbf{e}_0$:$\left\{(\mathbf{x}_0^u, \mathbf{d}_0)^{(i)}\right\}$
for t=1,..T **do**
  Estimate $\boldsymbol{\Sigma}_{xd}$ from $\mathbf{e}_t \rightarrow \hat{\boldsymbol{\Sigma}}_{xd}$
  Forwarding:
  $\boldsymbol{\epsilon}_t^{x(i)} \sim N_{p_x}(\mathbf{0}, \mathbf{I}_{p_x})$
  $\mathbf{x}_{t+1}^u = \omega_t(\mathbf{x}_t^{c(i)}, \boldsymbol{\epsilon}_t^{x(i)})$
  $\boldsymbol{\epsilon}_{t+1}^{d(i)} \sim N_{p_d}(\mathbf{0}, \mathbf{I}_{p_d})$
  $\mathbf{d}_{t+1}^{(i)} = \nu_{t+1}(\mathbf{x}_{t+1}^{u(i)}, \boldsymbol{\epsilon}_{t+1}^{d(i)})$
  $\mathbf{e}_{t+1} : \left\{(\mathbf{x}_{t+1}^u, \mathbf{d}_{t+1})^{(i)}\right\}$
end for

the ensemble Kalman filter is not suited for the current situation.

The randomized maximum likelihood filter will not be described in this paper, but the reader is referred to Omre and Myrseth [16]. The randomized maximum likelihood filter requires a challenging and time consuming optimization step of the likelihood and is also left out as a solution to the filtering step.

4.3 Particle Filter

The basic particle filter is a sequential Monte Carlo method based on point mass, or particle, representations of probability densities [17]. The idea is to generate a set of N_s random samples $\mathbf{x}_{0:t}^i$ for $i = 1, \ldots, N_s$, from a proposal distribution $q(\mathbf{x}_{0:t}|\mathbf{d}_{1:t})$, and to estimate the posterior density function by assigning each sample a corresponding weight. The posterior distribution at time step t, can be approximated as,

$$f(\mathbf{x}_{0:t}|\mathbf{d}_{1:t}) \approx \sum_{i=1}^{N_s} w_t^i \delta\left(\mathbf{x}_{0:t} - \mathbf{x}_{0:t}^i\right), \tag{38}$$

where $\left\{\mathbf{x}_{0:t}^i, w_t^i\right\}_{i=1}^{N_s}$, denotes a random measure, and the weights are normalized so that $\sum_{i=1}^{N_s} w_t^i = 1$. The weights are asserted by importance sampling [17],

$$w_t^i \propto \frac{f(\mathbf{x}_{0:t}^i|\mathbf{d}_{1:t})}{q(\mathbf{x}_{0:t}^i|\mathbf{d}_{1:t})}. \tag{39}$$

By choosing an importance density so that it factorizes to

$$q(\mathbf{x}_{0:t}|\mathbf{d}_{1:t}) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{d}_{1:t}) q(\mathbf{x}_{0:t-1}|\mathbf{d}_{1:t-1}), \tag{40}$$

it can be shown that [17],

$$w_t^i \propto w_{t-1}^i \frac{p(\mathbf{d}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{0:t-1}^i, \mathbf{d}_t)}. \tag{41}$$

Furthermore, by choosing $q(\mathbf{x}_t^i|\mathbf{x}_{0:t-1}^i, \mathbf{d}_t) = q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{d}_t)$, the weights can be sequentially found as,

$$w_t^i \propto w_{t-1}^i \frac{L(\mathbf{d}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}. \tag{42}$$

The most convenient choice for the importance distribution is the prior, then the weights reduce to,

$$w_t^i \propto w_{t-1}^i L(\mathbf{d}_t|\mathbf{x}_t^i).$$

One practical problem with the particle filter is known as *degeneracy*, that is after a few iterations all but one particle is of negligible weight. A good choice of importance density is crucial, but in many situation there is still a need of resampling. The full method for resampling will not be presented here, but the basic idea is to eliminate particles that have small weights and to concentrate on the particles with large weights. Resampling involves generating a new set $\{\mathbf{x}_t^{i*}\}_{i=1}^{N_s}$ by resampling N_s times from an approximate discrete representation of,

$$\begin{aligned} p(\mathbf{x}^*{}_t|d_{1:t}) &\approx \sum_{i=1}^{N_s} w_t^i \delta(\mathbf{x}_t^* - \mathbf{x}_t^i) \\ &= \frac{1}{N_s}. \end{aligned}$$

For further details and full algorithm the reader is referred to [17].

Let us now turn back to our case with the basis functions for precipitation and the advection field. Remember from section 3.6 that inactive functions are reset at the next update. Generating particles forward in time would demand that we somehow modify the particle filter, allowing for the classification of inactive functions when filtering the basis functions $\mathbf{c}_t$, $\mathbf{w}_t$ and $\mathbf{h}_t$, and not taking these forward in time when updating the advection, $\mathbf{u}_t$. Choosing the prior as proposal distribution in equation 42 is certainly a far too simple choice, requiring us to generate a vast number particles to filter a new satisfying image because of the high dimensionality of the model. In particular the initialization of the image, requires a very large number of particles before satisfying results are obtained. Other proposals could certainly perform better, but in addition there is the problem of degeneracy. Of the filters presented so far, the particle filter is certainly the most suited one, since it allows for highly nonlinear state space models [17].

4.4 Bayesian filtering, MCMC update

Applying any of the filters described in section 4.2 and 4.3, on the model specified in section 3, is not straight forward. As discussed there are several reasons for this. The likelihood from equation 20 model is highly non linear, having $N \times 4$ stochastic variables at each time step, N being the number of basis functions. The hidden Markov chain shown in figure 7, is rather complex since we have to update the advection holding the basis parameters constant, and vice versa. The situation would have been more comfortable if the basis variables could have been updated without dealing with a changing advection field. In addition to this we have the issue of a varying parameter space for $\mathbf{u}_t$.

4.4.1 Metropolis-Hastings algorithm

Before we introduce the filtering scheme using MCMC, the sampling technique must be specified. The most used MCMC updated scheme is perhaps the Metropolis-Hastings (MH) algorithm. Assume we want to explore the multivariate distribution,

$$\pi(\mathbf{x}) = \pi(x_1, \ldots, x_{n_{px}}), \tag{43}$$

and that both standard Monte Carlo sampling and numerical integration is unavailable. The normalization factor is often extremely difficult to compute, but the ability to generate a sample without knowing this constant of proportionality is a major virtue of the MH algorithm. The target distribution, $\pi(\mathbf{x})$, is assumed from Bayesian framework, proportional to the likelihood, $L(\mathbf{x})$, and the prior, $p(\mathbf{x})$,

$$\pi(\mathbf{x}) \propto L(\mathbf{x})p(\mathbf{x}).$$

The main idea is to construct a Markov chain that converges on $\pi(\mathbf{x})$. The MH algorithm uses a proposal density, $Q(\hat{\mathbf{x}}|\bar{\mathbf{x}})$, which can generate a proposed value, $\hat{\mathbf{x}}$, given the previous sample, $\bar{\mathbf{x}}$. Whether the proposed value is accepted or rejected as the new value, is decided by the acceptance probability,

$$\alpha = min\left\{1, \frac{\pi(\hat{\mathbf{x}})}{\pi(\bar{\mathbf{x}})}\frac{Q(\bar{\mathbf{x}}|\hat{\mathbf{x}})}{Q(\hat{\mathbf{x}}|\bar{\mathbf{x}})}\right\}. \tag{44}$$

By drawing a uniform variable u, on the interval $[0, 1]$, $\hat{\mathbf{x}}$ is accepted as the new value if u is smaller than α. Using a random walk proposal, the proposal distribution $Q(\hat{\mathbf{x}}|\bar{\mathbf{x}})$ is symmetric, and it is easy see from equation 44 that the probability for acceptance α, reduce to,

$$\begin{aligned}\alpha &= min\left\{1, \frac{\pi(\hat{\mathbf{x}})}{\pi(\bar{\mathbf{x}})}\right\} \\ &= min\left\{1, \frac{L(\hat{\mathbf{x}})p(\hat{\mathbf{x}})}{L(\bar{\mathbf{x}})p(\bar{\mathbf{x}})}\right\}.\end{aligned}$$

Sampling from one variable at the time, conditioned on the other parameters, is called single site MH algorithm. Algorithm 3 describes the procedure for a single site MH algorithm with a random walk proposal.

Algorithm 3 Metropolis-Hastings single site update

Initiate:
$\mathbf{x^0} = \left[x_1^0, ..., x_{n_{p_x}}^0\right]'$
for i=1 to N_s **do**
  for j=1 to n_{p_x} **do**
    $\hat{x}_j^{(i)} \sim N(x_j^{(i-1)}, a)$ {a is tuning parameter}
    $\hat{\mathbf{x}}_j^{(i)} = \left[x_1^{(i)}, \ldots, \hat{x}_j^{(i)}, \ldots, x_{n_{p_x}}^{(i)}\right]'$
    $\alpha_j^{(i)} = \frac{L(\hat{\mathbf{x}}_j^{(i)})p(\hat{\mathbf{x}}_j^{(i)})}{L(\mathbf{x}_j^{(i)})p(\mathbf{x}_j^{(i)})}$ {Acceptance probability}
    $u \sim U[0,1]$
    if $\alpha_{t_j} < u$ **then**
      $x_j^{(i)} = \hat{x}_{t_j}^{(i)}$
    else
      $x_j^{(i)} = x_j^{(i-1)}$
    end if
  end for
end for

4.4.2 MCMC filtering

The idea presented in [7] is quite simple. Assume we want to do an online sampling of the sequential target distribution $\pi(\mathbf{x}_{t+1}|\mathbf{d}_{t+1})$. Further assume that a MCMC run is available from time $0, ..., t$. By estimating $\pi(\mathbf{x}_{t+1}|\mathbf{d}_{0:t+1})$ as a discrete distribution, and then sampling from the joint distribution $\pi(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{d}_{t+1})$, via an empirical resampling of the previous MCMC run from the last time step, a highly efficient and powerful algorithm is obtained for online tracking. The joint posterior distribution $\pi(\mathbf{x_{t+1}}, \mathbf{x_t}|\mathbf{d_{1:t+1}})$ can easily be obtained via Bayes rule,

$$\begin{aligned}\pi(\mathbf{x_{t+1}}, \mathbf{x_t}|\mathbf{d_{1:t+1}}) &\propto \pi(\mathbf{x_{t+1}}, \mathbf{x_t}, \mathbf{d_{1:t}}, \mathbf{d_{t+1}}) \\ &\propto \pi(\mathbf{d_{t+1}}|\mathbf{x_{t+1}}, \mathbf{x_t}, \mathbf{d_{1:t}}) \cdot \pi(\mathbf{x_{t+1}}, \mathbf{x_t}, \mathbf{d_{1:t}}) \\ &\propto \mathbf{L}(\mathbf{d_{t+1}}|\mathbf{x_{t+1}}) \cdot \mathbf{p}(\mathbf{x_{t+1}}|\mathbf{x_t}) \cdot \pi(\mathbf{x_t}|\mathbf{d_{0:t}}). \end{aligned} \tag{45}$$

The resampling of the previous filtration is done empirically so that,

$$\pi(\mathbf{x_t}|\mathbf{d_{0:t}}) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \delta_{\mathbf{x}_t^{(i)}}(\mathbf{x}_t^{(i)}), \tag{46}$$

where N_s is the number of simulations after the MCMC run has converged. The algorithm for the MCMC scheme is given from algorithm 4.

Algorithm 4 Bayesian filtering with MCMC

Initiate:
$\mathbf{x}_0^{(1)}, ..., \mathbf{x}_0^{(N_s)} \sim f(\mathbf{x}_0|\mathbf{d}_0)$ {Initial distribution}
for t=0,...,T-1 **do**
  Sample from the joint distribution by MCMC using algorithm 3.
  $[\mathbf{x_{t+1}}, \mathbf{x_t}|\mathbf{d_{0:t+1}}] \sim \mathbf{L}(\mathbf{d_{t+1}}|\mathbf{x_{t+1}}) \cdot \mathbf{p}(\mathbf{x_{t+1}}|\mathbf{x_t}) \cdot \pi(\mathbf{x_t}|\mathbf{d_{0:t}})$
end for

The advantages with this approach over the importance sampling approach such as the particle filter, are that no weighting is required and degeneracy is avoided. The MCMC approach also deals very well with high dimensional and structured state space models [7]. However, remember that we have to update the advection $\mathbf{u}_t$ holding the basis functions $\mathbf{y}_t$ constant and vice versa. In the general case were $\mathbf{x}_t$ is the stochastic variable we want to sample from, the hierarchical structure is given in figure 2 in section 3.1. Our model, whose hierarchical model is specified in figure 7 in section 3.3.3, differs in structure, since we update $\mathbf{u}_t$ and $\mathbf{y}_t$ sequential holding one variable constant when updating the other. Therefore we must sample from both target distributions for $\pi(\mathbf{u}_{t+1}, \mathbf{u}_t|\mathbf{I}_0, \ldots, \mathbf{I}_{t+1})$ and $\pi(\mathbf{y}_{t+1}, \mathbf{y}_t|\mathbf{I}_0, \ldots, \mathbf{I}_{t+1})$. In addition to this, $\mathbf{u}_t$ differ in its dimension of the parameter space, making sampling from a joint distribution an even more intricate process. Finally, since everything happens online, it is difficult to verify burn in and implement a routine that discard realizations before convergence is achieved. Due to these issues we have chosen to make a simplification to algorithm 4.

An approximative solution is to hold $\mathbf{x}_t$ in $\pi(\mathbf{x}_{t+}, \mathbf{x_t}|\mathbf{d}_{0:t})$ constant, by setting it to a proper value. Assume previous run is available, $\mathbf{x}_t^{(i)}$ for $i = 1, \ldots, N_s$ and that the chain has converged so that the last realization is a true realization from $\pi(\mathbf{x}_t|\mathbf{I}_{0:t})$. By taking only the last value, $\mathbf{x}_t^{(N_s)}$ when updating next conditioned variable $\mathbf{x}_{t+1}|\mathbf{I}_{0:t+1}$, algorithm 4 reduces to algorithm 5. We use last realization of $\mathbf{x}_t$ instead of the mean for two reasons. First, since the model is not expected to be Gauss linear it could be multi heaped, indicating the mean being an improbable value. Secondly the sampling is done online, making it hard to implement a routine which cuts and uses only values after burn in has occurred.

The verification of this approximation is somewhat informal. The target distributions for the single and the joint sampler are,

$$\begin{aligned}\pi(\mathbf{x}_{t+1}|d_{1:t+1}) &\propto L(\mathbf{d}_{t+1}|\mathbf{x}_{t+1}) \cdot p(\mathbf{x}_{t+1}|\bar{\mathbf{x}}_t)\\ \pi(\mathbf{x}_{t+1}, \mathbf{x}_t|d_{1:t+1}) &\propto L(\mathbf{d}_{t+1}|\mathbf{x}_{t+1}) \cdot p(\mathbf{x}_{t+1}|\mathbf{x}_t) \cdot \pi(\mathbf{x}_t|\mathbf{d}_{1:t}).\end{aligned} \tag{47}$$

When performing a MCMC run in the joint sampler, the term $\pi(\mathbf{x}_t|\mathbf{d}_{1:t})$ will not affect

Algorithm 5 Approximative Bayesian filtering with MCMC

Initiate:
$\mathbf{x}_0^{(1)}, ..., \mathbf{x}_0^{(N_s)} \sim f(\mathbf{x}_0|\mathbf{d}_0)$ {Initial distribution}
for t=0,...,T-1 **do**
  $\bar{\mathbf{x}}_t = \mathbf{x}_t^{(N_s)}$
  Sample from single distribution by MCMC using algorithm 3.
  $[\mathbf{x_{t+1}}|\mathbf{d_{0:t+1}}] \sim L(\mathbf{d}_{t+1}|\mathbf{x}_{t+1}) \cdot p(\mathbf{x}_{t+1}|\bar{\mathbf{x}}_t)$
end for

the acceptance probability. Hence both target distributions reduce to the product of the likelihood $L(\mathbf{d}_{t+1}|\mathbf{x}_{t+1})$ and a slightly different prior distributions, $p(\mathbf{x}_{t+1}|\cdot)$. Which of the factors that dominates is due to the amount of information they contain relative to the other. In many situations the likelihood contains enough data to say it is the dominating factor. We will assume that this is also the case in our situation, since the likelihood contains M^2 data points, and assuming the variance Σ_I being relative small. The likelihood will then dominate in both distributions, and the sampling from $\pi(\mathbf{x}_{t+1}|\mathbf{I}_{0:t+1})$ will yield approximately the same results as sampling from $\pi(\mathbf{x}_{t+1}, \mathbf{x}_t|\mathbf{I}_{0:t+1})$. In addition to the likelihood being the dominant factor, the priors are expected to be of the same family, only having a different mean value. Sampling from the approximative distribution is then assumed sufficient. Intuitively this will underestimate the variance, since we do not sample from the entire sample space but only sample with respect a new value with respect to the last realization of previous run. The mean however will hopefully be close to the true value since each parameter are adjusted according to the likelihood.

We can soon define an update scheme for the filtered and forecast values of $\mathbf{R}$. Before we do this, we must derive the posterior distributions for the basis functions, $\mathbf{y}_t$, and advection, $\mathbf{u}_t$. The deviation of the posterior distributions is simple as our model is a two level hierarchical model, visualized in figure 7 in section 3.3.3. When updating the advection $\mathbf{u}_t$, treat the basis parameters $\mathbf{y}_t$ and previous advection $\mathbf{u_{t-1}}$ as constants, $\bar{\mathbf{y}}_{t-1}$ and $\bar{\mathbf{u}}_t$. Hence we get for the advection,

$$\begin{aligned}\pi(\mathbf{u}_t|\mathbf{I}_{0:t}) &\propto \pi(\mathbf{u}_t, \mathbf{I}_{0:t}) \\ &\propto \pi(\mathbf{I}_{0:t}|\mathbf{u}_t)\pi(\mathbf{u}_t) \\ &\propto L(\mathbf{I}_t|\bar{\mathbf{c}}_{t-1} + \delta t \cdot \mathbf{u}_t, \bar{\mathbf{w}}_{t-1}, \bar{\mathbf{h}}_{t-1}, \bar{\mathbf{u}}_{t-1})p(\mathbf{u}_t|\bar{\mathbf{u}}_{t-1}). \end{aligned} \tag{48}$$

Deriving the posterior distributions of the basis functions $\mathbf{y}_t$, is similar, updating the last variable holding everything else constant,

$$\begin{aligned}\pi(\mathbf{y}_t|\mathbf{I}_{0:t}) &\propto \pi(\mathbf{y}_t, \mathbf{I}_{0:t}) \\ &= \pi(\mathbf{I}_{0:t}|\mathbf{y}_t)\pi(\mathbf{y}_t) \\ &\propto L(\mathbf{I}_t|\mathbf{y}_t)p(\mathbf{y}_t|\bar{\mathbf{y}}_{t-1}, \bar{\mathbf{u}}_t).\end{aligned}$$

If we write this out for each basis parameter we get,

$$\pi(\mathbf{c}_t|\mathbf{I}_{0:t}) \propto L(\mathbf{I}_t|\mathbf{c}_t, \mathbf{w}_t, \mathbf{h}_t)p(\mathbf{c}_t|\bar{\mathbf{c}}_{t-1}, \bar{\mathbf{u}}_t) \tag{49}$$

$$\pi(\mathbf{w}_t|\mathbf{I}_{0:t}) \propto L(\mathbf{I}_t|\mathbf{c}_t, \mathbf{w}_t, \mathbf{h}_t)p(\mathbf{w}_t|\bar{\mathbf{w}}_{t-1}) \tag{50}$$

$$\pi(\mathbf{h}_t|\mathbf{I}_{0:t}) \propto L(\mathbf{I}_t|\mathbf{c}_t, \mathbf{w}_t, \mathbf{h}_t)p(\mathbf{h}_t|\bar{\mathbf{h}}_{t-1}). \tag{51}$$

Using the single site update algorithm we propose to change only one value at the time, that is you sample one variable conditioned on everything else. This corresponds to the full conditional distributions. If we had been able to sample from these directly, it can be shown that the acceptance probability would always be of value 1 [18]. This is called the Gibbs sampler, and is the mos widely used version of the MH algorithm. However, our full conditional distributions can not be sampled directly from, and the acceptance probability must be calculated at each step.

For any N dimensional variable $\mathbf{z}_t$, let,

$$\mathbf{z}_t^{-n} = z_{t_1}, \ldots, z_{t_{n-1}}, z_{t_{n+1}}, \ldots, z_{t_N}.$$

The full conditional distributions for a basis function n, then reduce to,

$$\begin{aligned}\pi(y_{t_n}|\mathbf{I}_{0:t}, \mathbf{y}_t^{-n}) &\propto \pi(\mathbf{I}_{0:t}, \mathbf{y}_t) \\ &\propto L(\mathbf{I}_t|\mathbf{y}_t) \cdot p(y_{t_n}|\bar{y}_{t-1_n}, \bar{u}_{t_n}).\end{aligned} \tag{52}$$

Here the distribution of the prior $p(\cdot)$, of each y_{t_n} will be classified whether the previous basis function y_{t-1_n}, is classified as active or inactive. If the previous basis function i passive, the prior is set to the initial prior, given from equation 25 in section 3.5. In fact, all priors for inactive basis functions are proper or improper uniform distributions. In this case, it can be seen directly from equation 52 that the resulting posterior distribution is proportional to the likelihood. If the previous basis function is classified as active, that is $A_{t-1_n} = 1$, the multi normal priors from section 3.3, equation 12, 13 and 14 are chosen. The full conditional distribution for one center, c_{t_n}, with corresponding active y_{t-1_n}, given everything else, then becomes,

$$\pi(c_{t_n}|\mathbf{I}_{0:t}, \mathbf{c}_t^{-n}, \mathbf{h}_t, \mathbf{w}_t, A_{t-1_n} = 1) \propto L(\mathbf{I}_t|\mathbf{c}_t, \mathbf{h}_t, \mathbf{w}_t) \cdot \exp\left\{-\frac{1}{2}\frac{(c_{t_n} - \bar{c}_{t-1_n} - \delta_t \cdot \bar{u}_{t_n})^2}{\sigma_c^2}\right\}. \tag{53}$$

As for the centers, the full conditional distribution for the height function h_{t_n} with corresponding active basis function y_{t-1_n}, becomes,

$$\pi(h_{t_n}|\mathbf{I}_{0:t}, \mathbf{h}_t^{-n}, \mathbf{c}_t, \mathbf{w}_t, A_{t-1_n} = 1) \propto L(\mathbf{I}_t|\mathbf{c}_t, \mathbf{h}_t, \mathbf{w}_t) \cdot \exp\left\{-\frac{1}{2}\frac{\left(h_{t_n} - \bar{h}_{t-1_n}\right)^2}{\sigma_h^2}\right\}. \tag{54}$$

By the same argument, the full conditional distribution for the weight function w_{t_n} with corresponding active basis function y_{t-1_n}, becomes,

$$\pi(w_{t_n}|\mathbf{I}_{0:t}, \mathbf{w}_t^{-n}, \mathbf{c}_t, \mathbf{h}_t, A_{t-1_n} = 1) \propto L(\mathbf{I}_t|\mathbf{c}_t, \mathbf{h}_t, \mathbf{w}_t) \cdot \exp\left\{-\frac{1}{2}\frac{(w_{t_n} - \bar{w}_{t-1_n})^2}{\sigma_w^2}\right\}. \quad (55)$$

If the previous corresponding basis function is inactive, that is $A_{t-1_n} = 0$, all priors are uniform, and it can be seen directly from equation 52 that all full conditional distributions simply reduce to the likelihood,

$$\pi(y_{t_n}|\mathbf{I}_{0:t}, \mathbf{y}_t^{-n}, A_{t-1_n} = 0) \propto \pi(\mathbf{I}_{0:t}, \mathbf{y}_t) \propto L(\mathbf{I}_t|\mathbf{y}_t). \quad (56)$$

The full conditional distribution for the advection follow the same argument,

$$\pi(u_{t+1_n}|\mathbf{I}_{0:t+1}, \mathbf{u}_{t+1}^{-n}) \propto L(\mathbf{I}_{t+1}|\bar{\mathbf{c}}_t + \delta t \cdot \mathbf{u}_{t+1}, \bar{\mathbf{h}}_t, \bar{\mathbf{w}}_t) \cdot p(\mathbf{u}_{t+1}|\bar{\mathbf{u}}_t), \quad (57)$$

where the likelihood has been altered by substituting $\bar{\mathbf{c}}_t + \delta t \cdot \mathbf{u}_t$ for $\bar{\mathbf{c}}_{t+1}$ since the advection has to be updated before the basis parameters can be obtained. The full conditional distribution then becomes,

$$\pi(u_{t_n}|\mathbf{I}_{0:t}, \mathbf{u}_t^{-n}, A_{t_{n-1}} = 1) \propto \quad (58)$$

$$L^*(\mathbf{I}_t|\bar{\mathbf{c}}_{t-1} + \delta t \cdot \mathbf{u}_t, \bar{\mathbf{h}}_t, \bar{\mathbf{w}}_t) \cdot \exp\left\{-\frac{1}{2}(\mathbf{u}_t - \bar{\mathbf{u}}_{t-1})'\mathbf{\Sigma}_{u_t}(\mathbf{u}_t - \bar{\mathbf{u}}_{t-1})\right\}. \quad (59)$$

L^* indicates the likelihood of only active basis functions. The design of $\mathbf{\Sigma}_{u_t}$ and asserting values for $\bar{u}_{t_j}$ when the basis function is recently activated, is done according to the framework presented in section 3.6. The full algorithm for our update scheme is given as algorithm 6.

When realizations of basis parameters with corresponding advection are available, we can easily obtain the filtered image and forecasts. We use the standard estimators for mean and prediction intervals. The mean of the filtered and predicted image is then,

$$E[\mathbf{R}_t^c] = \frac{1}{N_s} \cdot \sum_{i=1}^{N_s} \mathbf{R}_t(\mathbf{c}_t^{(i)}, \mathbf{w}_t^{(i)}, \mathbf{h}_t^{(i)}) \quad (60)$$

$$E[\mathbf{R}_{t+\delta t}^u] = \frac{1}{N_s} \cdot \sum_{i=1}^{N_s} \mathbf{R}_{t+\delta t}(\mathbf{c}_t^{(i)} + \delta t \bar{\mathbf{u}}_t, \mathbf{w}_t^{(i)}, \mathbf{h}_t^{(i)}). \quad (61)$$

The estimator for the precipitation level to exceed a specified value z at given location $\mathbf{x}$ is given as,

$$P(R_t(\mathbf{x}) > z) = \frac{1}{N_s} \cdot \sum_{i=1}^{N_s} \delta(R_t^{(i)}(\mathbf{x}) > z)). \quad (62)$$

When updating c_{t+1_n}, and the previous center is inactive, $A_{t_n} = 0$, c_{t+1_n} has equal probability to be located anywhere on the grid. We exploit this by starting the MCMC run by asserting the starting value for the previous inactive centers at the most underestimated location, $\mathbf{x} \in \mathbf{X}$, given the expected value of the previous forecast of,

$$c_{t+1_n}^{(0)} = max_{\mathbf{x}} \left(\mathbf{I}_{t+1} - E[R_{t+1}^{u}(\mathbf{X})] \right). \tag{63}$$

This helps the MCMC run to converge faster, and quickly locating any new rain showers that might have entered the system.

Algorithm 6 MCMC update scheme for full model

Set Advection at first image to zero:
$\mathbf{u}_0^{(i)} = 0$ for i=1,...,N_{sim}
Initialize first image:
for i=1 to N_{sim} **do**
  Sample $\mathbf{c}_0, \mathbf{w}_0, \mathbf{h}_0$ using MCMC from algorithm 3:
  $\pi(\mathbf{c}_0^{(i)}, \mathbf{w}_0^{(i)}, \mathbf{h}_0^{(i)}) \sim L(\mathbf{c}_0^{(i)}, \mathbf{w}_0^{(i)}, \mathbf{h}_0^{(i)})$
end for
$\bar{\mathbf{A}}_0 \leftarrow getAct(\bar{\mathbf{c}}_0, \bar{\mathbf{w}}_0, \bar{\mathbf{h}}_0)$ {Find Active basis functions at t_0}
for t=1 to T **do**
  $\bar{\mathbf{u}}_{t-1} = \mathbf{u}_{t-1}^{(N_{sim})}$ {Treat previous values as constants.}
  $\bar{\mathbf{c}}_{t-1} = \mathbf{c}_{t-1}^{(N_{sim})}$, $\bar{\mathbf{w}}_{t-1} = \bar{\mathbf{w}}_{t-1}^{(N_{sim})}$, $\bar{\mathbf{h}}_{t-1} = \mathbf{h}_{t-1}^{(N_{sim})}$
  Update advection field $\mathbf{u}_t$.
  $\mathbf{u}_t^{(0)} = \bar{\mathbf{u}}_{t-1}$ {Initialize advection field}
  for $i = 1$ to N_{sim} **do**
    Sample $\mathbf{u}_t^{(i)}$ using MCMC from algorithm 3:
    $\pi(\mathbf{u}_t^{(i)}) \sim L(\mathbf{I}_t|\mathbf{u}_t^{(i)} + \bar{\mathbf{c}}_{t-1}, \bar{\mathbf{w}}_{t-1}, \bar{\mathbf{h}}_{t-1}, \mathbf{A}_{t-1}) p(\mathbf{u}_t^{(i)}|\bar{\mathbf{u}}_{t-1}, \mathbf{A}_{t-1})$
  end for
  Update basis functions $\mathbf{c}_t$, $\mathbf{w}_t$ and $\mathbf{h}_t$.
  $\bar{\mathbf{u}}_t = \mathbf{u}_t^{(N_{sim})}$ {Hold the advection field constant using only last realization}
  $\mathbf{c}_t^{(0)} = \mathbf{c}_{t_1}^{(N_{sim})}$, $\mathbf{w}_t^{(0)} = \mathbf{w}_{t_1}^{(N_{sim})}$, $\mathbf{h}_t^{(0)} = \mathbf{h}_{t_1}^{(N_{sim})}$ {Initialize basis functions}
  for $i = 1$ to N_{sim} **do**
    Sample $\mathbf{c}_t, \mathbf{w}_t, \mathbf{h}_t$ using MCMC from algorithm 3:
    $\pi(\mathbf{c}_t^{(i)}, \mathbf{w}_t^{(i)}, \mathbf{h}_t^{(i)}) \sim L(\mathbf{I}_t|\mathbf{c}_t^{(i)}, \mathbf{w}_t^{(i)}, \mathbf{h}_t^{(i)}, \bar{\mathbf{A}}_{t-1}) p(\mathbf{c}_t^{(i)}, \mathbf{h}_t^{(i)}, \mathbf{w}_t^{(i)}|\bar{\mathbf{c}}_{t-1}, \bar{\mathbf{w}}_{t-1}, \bar{\mathbf{h}}_{t-1}, \bar{\mathbf{u}}_t, \bar{\mathbf{A}}_{t-1})$
  end for
  $\bar{\mathbf{A}}_t \leftarrow getAct(\bar{\mathbf{c}}_t, \bar{\mathbf{w}}_t, \bar{\mathbf{h}}_t)$ {Find Active basis functions}
  if $\bar{A}_{t_j} = 1$ and $\bar{A}_{t-1_j} = 0$, for $j = 1, \ldots, N$ **then**
    $\bar{u}_{t_j} = mean(\bar{\mathbf{u}}_t)$ {Set advection value for new active centers to mean value.}
  else
    if $\bar{A}_{t_j} = 0$ **then**
      $\bar{u}_{t_j} = 0$ {If center is inactive, set advection to zero}
    end if
  end if
  Filtering
  $\bar{\mathbf{R}}_t^{(i)} = \mathbf{R}(\mathbf{c}_t^{(i)}, \mathbf{w}_t^{(i)}, \mathbf{h}_t^{(i)})$
  Forecasting
  $\hat{\mathbf{R}}_{t+1}^{(i)} = \mathbf{R}(\mathbf{c}_t^{(i)} + \delta t \bar{\mathbf{u}}_t, \mathbf{w}_t^{(i)}, \mathbf{h}_t^{(i)})$
end for

5 Result

5.1 Synthetic data

5.1.1 Initialization

To justify the model, we first start to verify it by using synthetic data generated from the model itself. This has the advantage that we can inference about known parameters to see if the model is operating correctly. After initializing the first image with corresponding advection, proper values for the covariance matrices were used to simulate the model forward in time. After simulating basis functions, random Gaussian noise was added over the entire grid. The lower critical limit for the precipitation field is chosen relatively large, so that the random noise added after the simulation of the basis functions does not exceed this limit. The parameters for the synthetic data is shown in table 1. To ensure that we infer about a satisfying number of basis functions, the number of basis functions used to generate the data is smaller than the number used to infer about the model. The standard deviation of the advection σ_u, is chosen relatively large to test the model flexibility when the advection might move rather rapidly.

Table 1: Parameters for synthetic data

Parameter	symbol	Value
Number of basis function	N	24
Number of simulated basis functions	N_b	16
Number of simulations	N_s	6000
Grid dimension	M	50
Advection correlation parameter	a	0.005
Standard deviation of advection	σ_u	1
Standard deviation of centers	σ_c	0.2
Standard deviation of radii	σ_w	0.1
Standard deviation of intensities	σ_h	0.1
Radar error	σ_I	0.1
Critical lower value for radii	$\mathbf{w_{crit}}$	0.4
Critical lower value for precipitation	$\mathbf{R_{crit}}$	0.5

The model was initialized at time t_0, and simulated sequentially seven steps forward in time. In the synthetic data case, only the last realization is visualized to justify our approximative MCMC scheme.

5.1.2 Filtering

The results for the filtering procedure for the six first time steps can be seen on the right side of figure 9. The left hand side of the figure shows the simulated precipitation

field together with added random noise. As expected the filtered images are very similar to the "radar" images, removes the added Gaussian noise, and smoothen the observed images. The top of figure 9 shows the initialized image, where the advection is not yet calculated. The images at time t_1 shows the filtered image where the advection is updated for the first time. Even though t_0 and t_1 are where the initialization is performed, they both yield good results. Observing the colorbars, we see that the intensities are also of approximately same magnitude. The reason for the filtered images to recreate the radar image so good, is that the elements of the variance matrix for the radar error $\mathbf{\Sigma}_I$, is relatively small, indicating that the radar image contains very accurate information. Another factor is the relative fine-meshed grid, providing enough calculations to provide sufficient information about the basis functions.

The advection can be found after initializing the first image, and the resulting advection is about of same magnitude and direction as the generated one. Observing the images for t_2 we can see that the "real" advection is somewhat of different direction at various geographical locations. Most of the vectors point to north east, but the vector located at the center furthest to west is pointing almost vertically. This is approximately identical to the filtered image, and we conclude that the filtering procedure is satisfying in this simple synthetic case.

5.1.3 Forecasting

After the initialization of the first image, and updating the advection field at time 1, the forecast prediction can be found for time 2. One step forecast predictions for time 2,3 and 4, compared with the true underlying image without the radar measurement noise, are shown in figure 10. The advection vectors are remain the same as for the previous filtered image since they are not expected to change. The forecast for t_2 looks very satisfying, due to the fact there is little growth or decay from t_1 to t_2, and that the direction of the advection stays approximately the same. The intensity of the forecast at t_2 compared to the real intensity seems is a bit high, but this probably the result of $\mathbf{w_{crit}}$ being too small, resulting in a blown up value at the location where the intensity is close to 4. The forecast at t_3 is similar in shape to the corresponding real image. However, we see that the advection has had a rapid change from t_2 to t_3, resulting in the forecast to move to far east. This is one weakness for this model, that it can not take into account a rapid change in advection when the next forecast is yielded. At t_4 we see that the change in advection has been adjusted, and that the location of the centers look good. However, there has been some decay in one of the western basis functions, and some growth of some of those to north east of the system, from t_3 to t_4. This can not be taken into account in the forecast, and the forecast will always be similar in shape compared to the previous filtered image.

Figure 11 shows the forecasts at time 5,6 and 7, but now the only available radar image is from time 4. This yields a one, two and three step prediction. The images to the left shows the corresponding true images at the given time. The advection used to forecast

all three images is found from the filtering at time 4. And since this, along with updating the basis parameters, are not adjusted for as time passes, the forecasts are increasingly inaccurate. The one step forecast is as expected the best one, and approximately recreates the true image, but the last to yields rather poor forecasts. One can see from the true images that the advection changes some between time 4 and 5. This explains some of the inaccuracy of the forecasts, since the advection used is quite different from the "real one". If this is not the case, the performance of the forecasting is expected to improve. However, even the two step forecast for t_7 gives us reliable information that the system has moved north and that it is about to leave the chosen geographical area.

From this synthetic case we can summarize that the filtering performs well, and handles a quite rapidly shifting advection field. The one step forecasts also yield good results, but forecast on further time intervals might yield mixed results, depending on the rate at which the advection change.

5.1.4 Realization and convergence of sampled parameters

We will now look at the convergence of precipitation levels at the grid locations, basis functions and advection. Figure 12 shows the realizations from the MCMC algorithm of the precipitation levels for the filtering at t_0. The figure only shows realizations for every tenth grid location to make the figure more surveyable. As seen from figure 12 the values converge quickly, being approximately normal. However, the situation for the underlying basis functions and advection are not as stable as for the resulting precipitation field.

Figure 13 shows the realizations of center coordinates $c_{0_{n_x}}$ at top, and $c_{0_{n_y}}$ at bottom, for basis functions n labeled as active at the last realization. As seen from the figure most of the center coordinates converge quickly, except from two. These two centers seem to randomly wander about until they "settle in" at about 5000 iterations. The situation for the advection is shown in figure 14, $u_{0_{n_x}}$ at top, and $u_{0_{n_y}}$ at bottom, is even more complex. Most realizations seem to converge about one single value, but some do not. This might be the result of their associating intensity h_{t_n} being of a small value, not yielding large effect to the likelihood when updating the specified advection u_{t_n}. This seems reasonable as the realizations that do not converge do not seem to affect the realizations of the precipitation field, as seen from figure 12. Figure 15 shows the number of basis functions classified as active at each realization at t_0, starting at 10, and having a maximum value of 22. The starting value 10 is not chosen, but a result of the random initialization of the MCMC run at time t_0. It is clear that the number of active basis functions does not converge to one single value. The run converges to values spanning from 18 to 22, which is below the true number, 24. It is reasonable that the filtered image is not able to recreate the full complexity of the true image as we have added some uncertainty in the radar image. The number of active basis functions at time t decides the parameter space of $\mathbf{u}_{t+1}$. This in addition to the chaotic behavior of the other underlying variables, justifies our approximative MCMC scheme from section 4.4.2, to use only the last realization to update the next parameter.

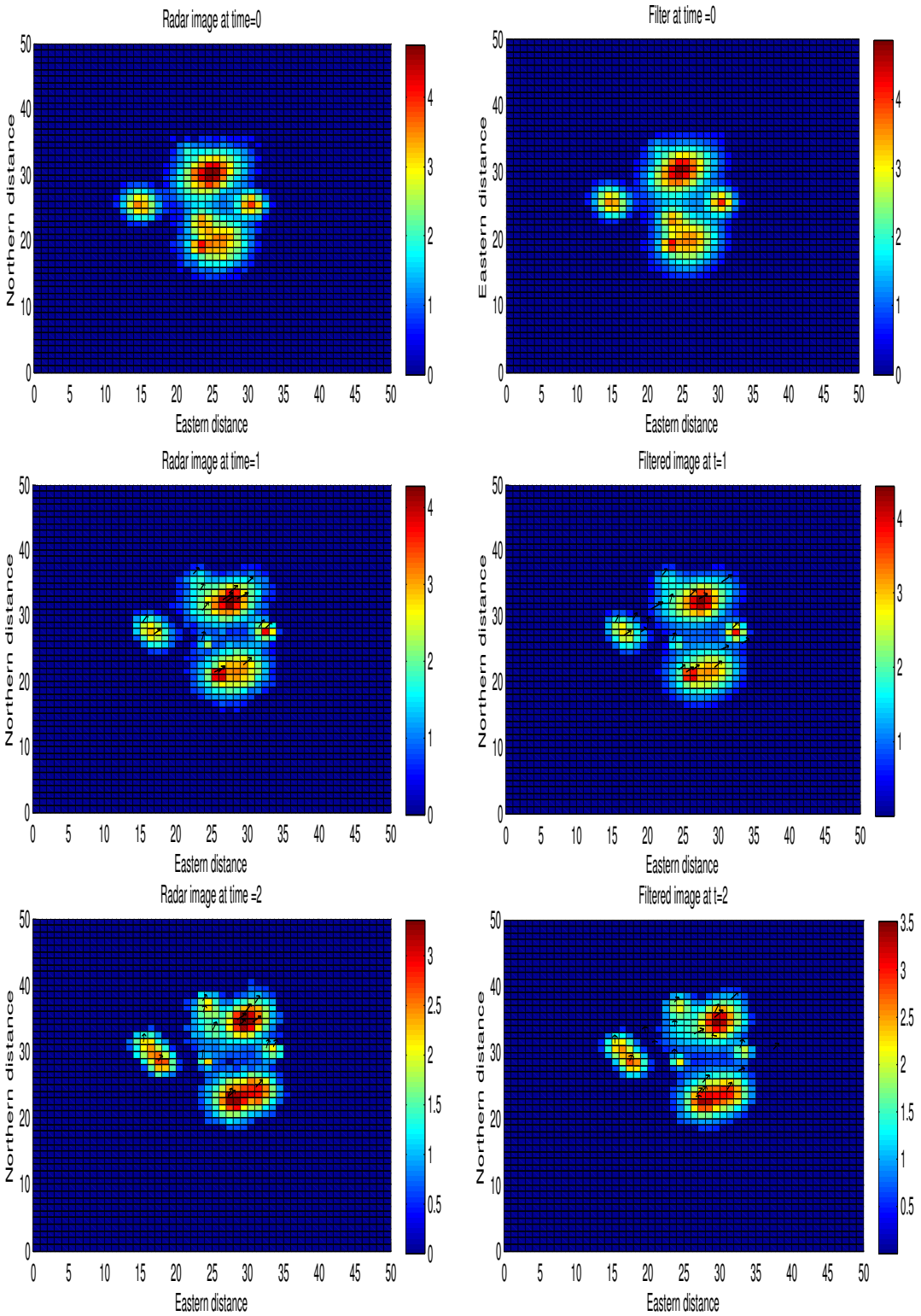


Figure 9: Radar images(left) with corresponding filtered images(right) for time t_0, t_1 and t_2. At time t_0 the advection is not yet initialized, but for time t_1 and t_2 the advection is of same magnitude and direction as the "real" advection. All three images are very similar in shape, and the "real" colorbars are showing about the same intensities as those from the filtered images.

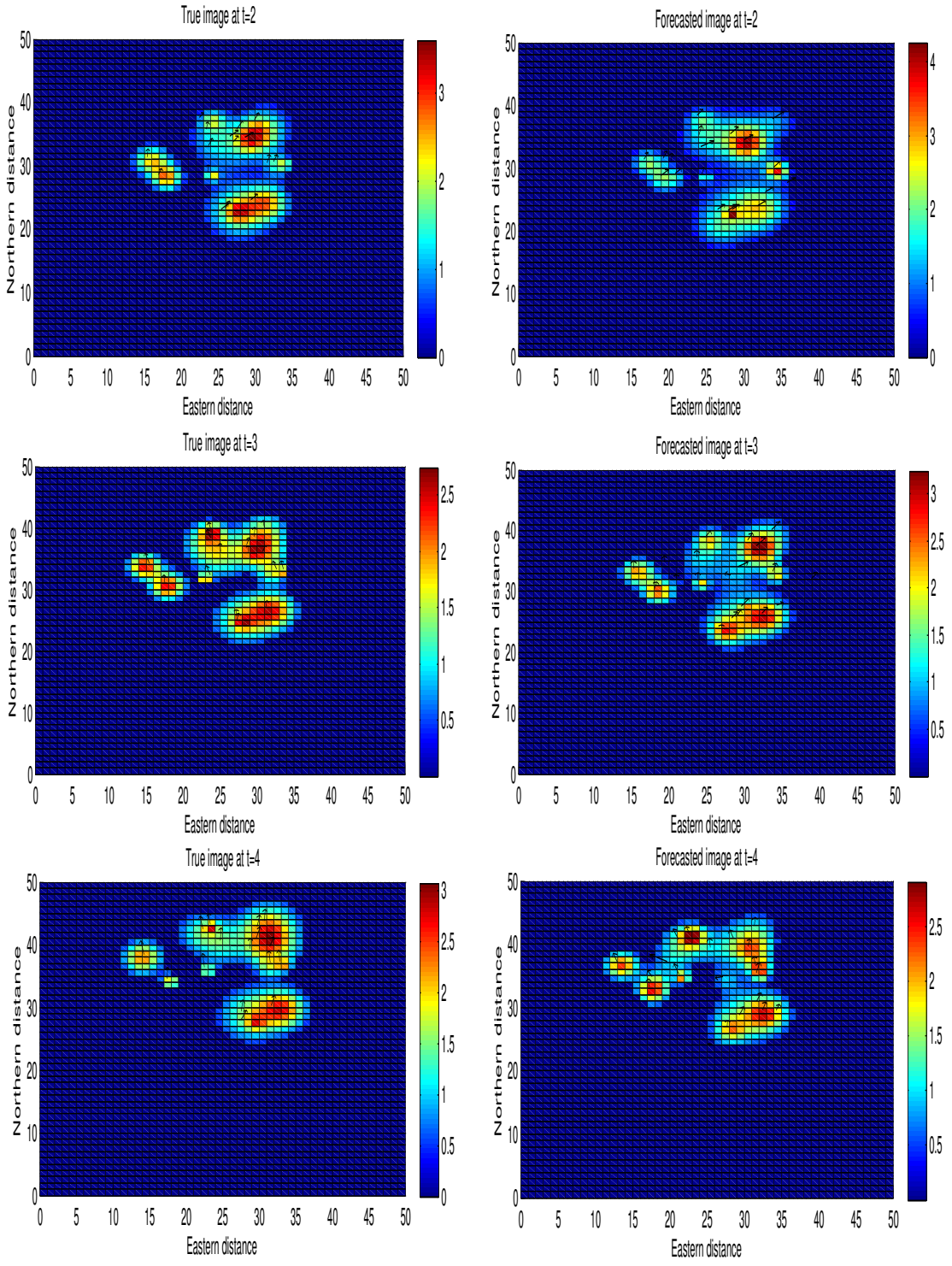


Figure 10: True images(left) and one step forecast images(right) at t_2, t_3 and t_4. The forecasts for t_2 and t_3 yield very good results as the precipitation fields have little changes between the updates. The forecast for t_4 is also satisfying, but due to some rapid change in advection and the amount of precipitation between t_3 and t_4 it is not as accurate as the two previous forecasts. The intensities of the filtered images, indicated by the colorbars, differ some compared to the "real" intensities because the distance from the centers to the closest pixels have changed slightly compared to the filtered ones.

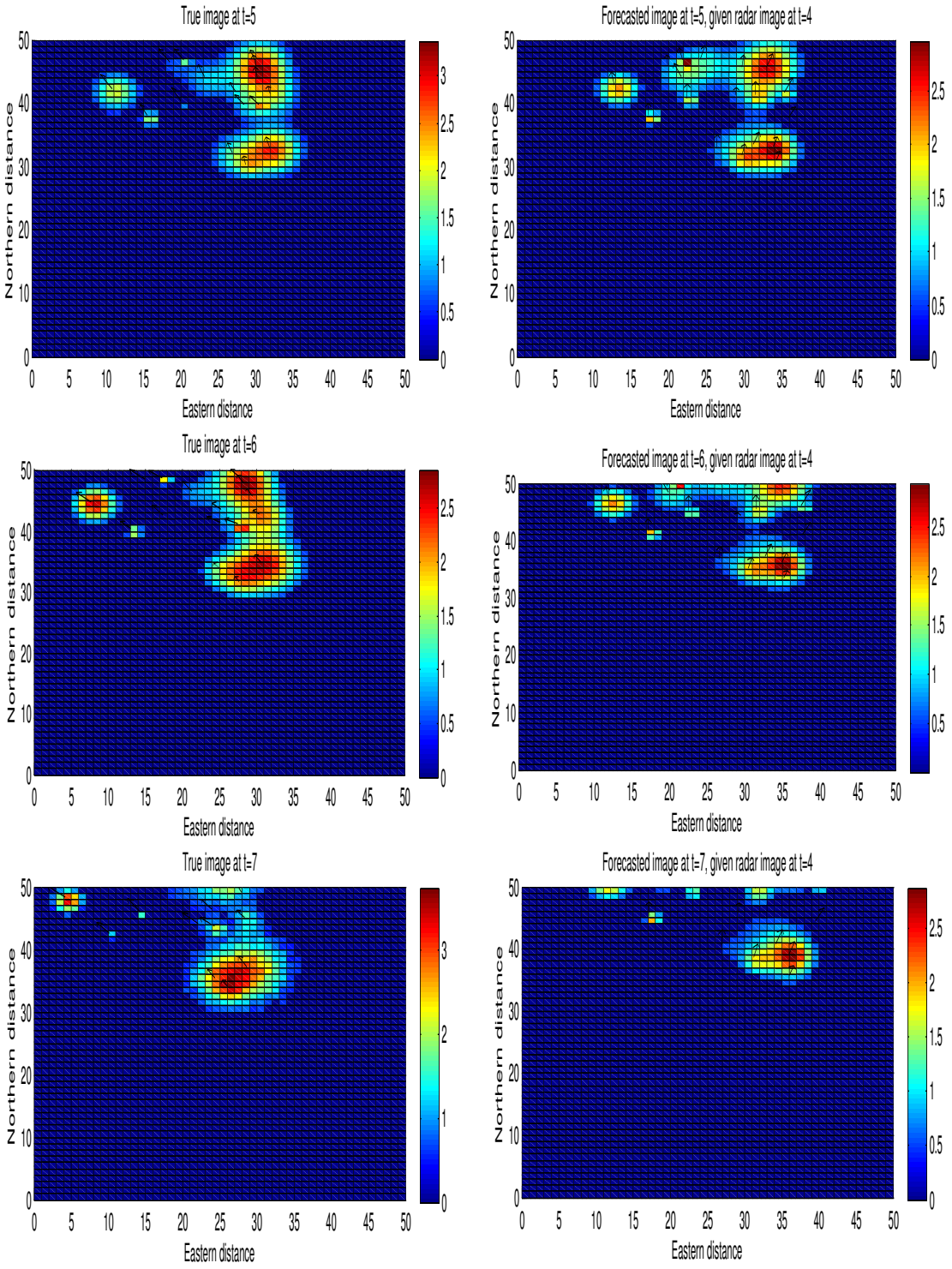


Figure 11: True images(left) and one, two and three step forecasts(right) at t_5, t_6 and t_7. All forecasts are conditioned on the "radar" image at t_3, and the advection is held constant for all three updates. The one step forecast at t_5 yield best result, and forecasts for t_6 and t_7 are increasingly inaccurate. However, both t_5 and t_6 yield approximately results as to how the predicted precipitation field will be. Some of the inaccuracy are ascribed to the rapid change in the advection field between t_4 and t_5. Also there is some growth and decay in the real images at time steps t_5, t_6 and t_7 which can not be accounted for.

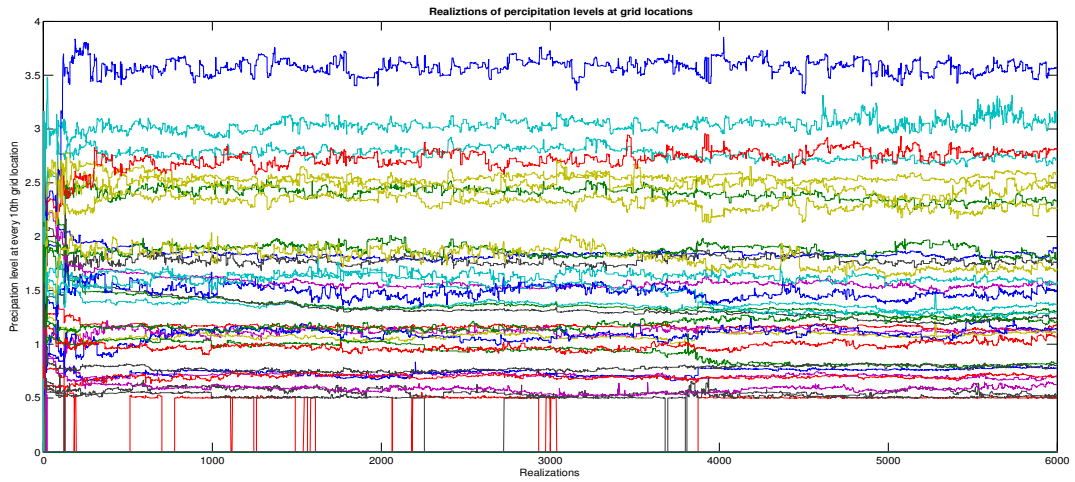


Figure 12: Realizations of precipitation levels at every 10th grid location, for time t_0. All values converge quickly, indicating a burn-in of less than 500 realizations. The values close to zero have discrete jumps from zero to a value close to the critical lower value $\mathbf{W_{crit}}$, set to 0.5 mm. pr. hour.

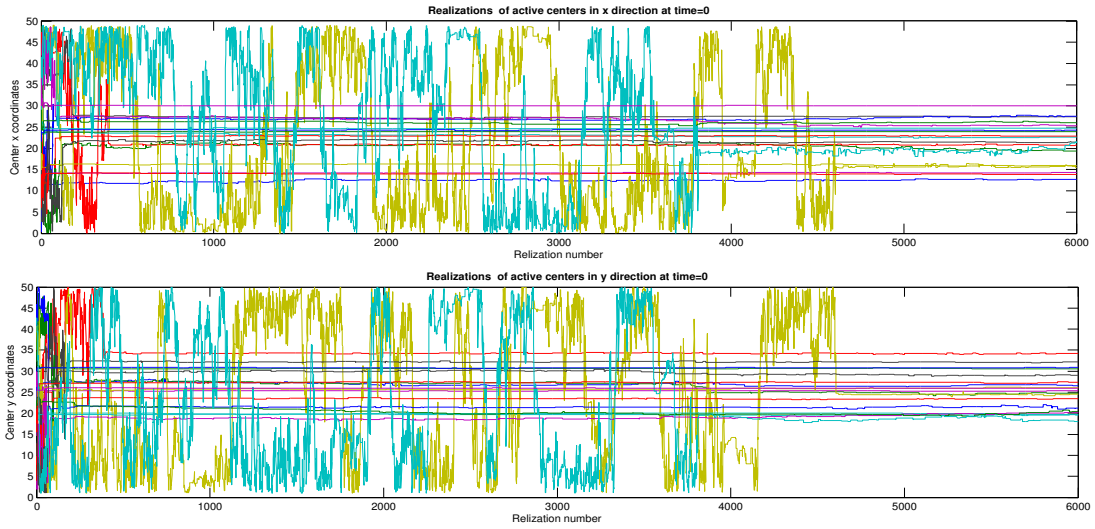


Figure 13: Realizations of coordinates of centers labeled as active at last realization at initial time, upper figure x coordinates, bottom figure y coordinates. Most centers converge quickly, but two centers performs a random walk over the legal area, until they finally find their place at about 4500 iterations.

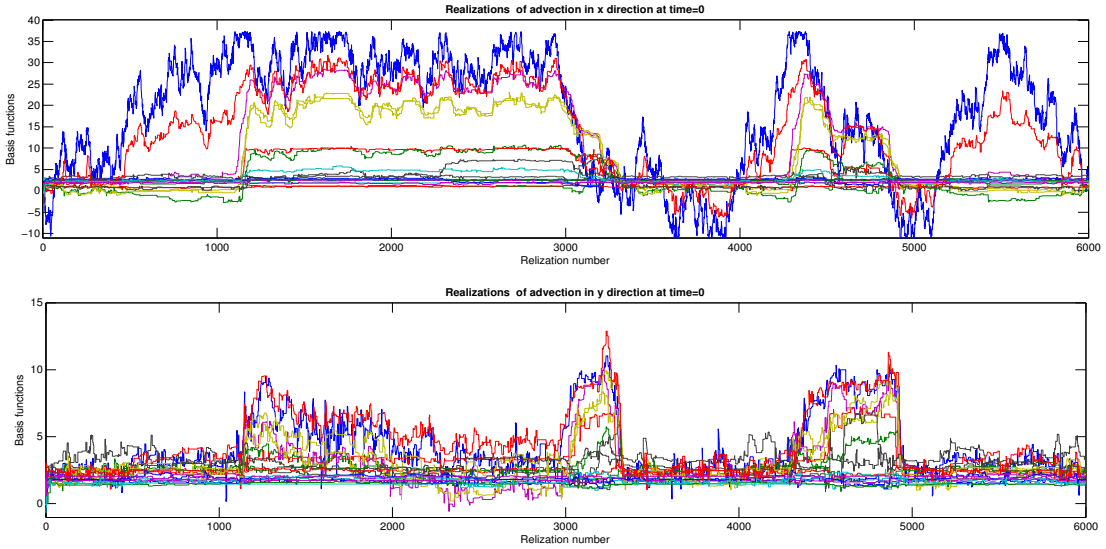


Figure 14: Realizations of advection at active centers at time t_1, upper figure x direction, bottom figure y direction. For many of the basis functions it is hard to decide convergence, illustrating the rather chaotic nature of the underlying distributions.

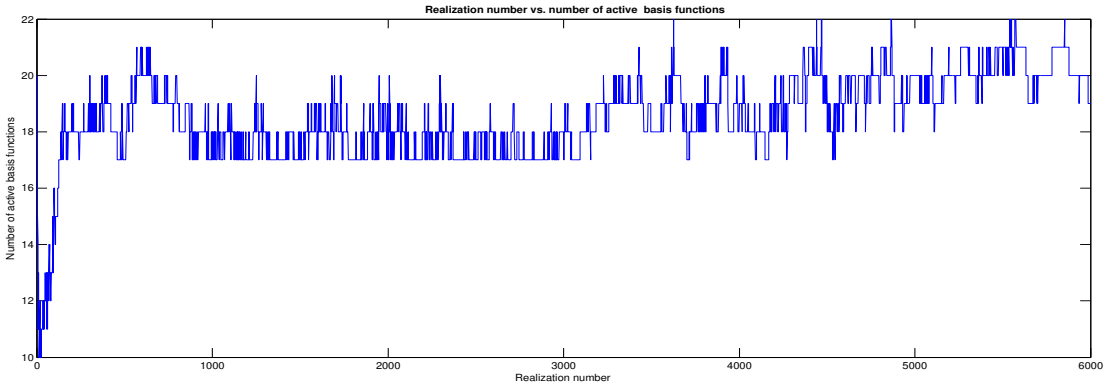


Figure 15: The classification of each basis function as active or passive at each realization at t_0. The initialization values for the basis functions are chosen at random, and this gives ten active basis functions at first realization. The synthetic precipitation field consists of 24 basis functions, and the realizations of the filtered image tends to have between 18 and 22 after the MCMC run seems to have converged. The run never converges to a single value, and this is one of the reasons why it is hard to sample from the entire sample space of the advection $\mathbf{u}_t$, as its parameter space varies.

5.2 Real data from the Rissa radar in Sør Trøndelag

Testing the model on real data images is more challenging because of explicit factors and uncertainties in the variances and correlation used in our model. We are not capable of forecasting growth and decay of new rain showers, but we are able to deal with this in the filtering procedure.The data are measured at 06.03.2006, providing a time series spanning from 12:00 to 18:00, and are gathered by the Rissa radar located in Sør Trøndelag, Norway. Figure 16 shows the geographical area of which we have chosen to evaluate, with the arrow pointing at Trondheim. Trondheim is the largest city in the area and and we will explore a two hour prediction from an image series where precipitation is moving towards the city. It should be mentioned that the radar image has been modified so that that the image used, only contains every fourth pixel of the original grid in each direction. The cell coordinates for Trondheim is $[28, 7]$ for the modified grid, as seen from figure 16 where the shown axes are scaled to fit the modified grid. The values used for each pixel at the unmodified grid is the mean value of the surrounding pixels and the pixel itself. This is done in order to shorten the run time in the MCMC run. In a full scale implementation all data point could be used, but this might severely increase the run time. The parameters chosen are shown in table 2. The images occur in a time difference of one hour, which might be a unfortunate since the model is based on a short time frame. Probably an occurrence sequence of 15 minutes would be ideal, but such data were not available at the time. All intensities are given as millimeter precipitation pr. hour.

Table 2: Parameters for real data

Parameter	symbol	Value
Number of basis function	N	25
Number of simulations	N_s	15000
Grid size	M	38
Correlation parameter	a	0.005
Advection standard deviation	σ_u	1
Center standard deviation	σ_c	1
Width standard deviation	σ_w	4
Height standard deviation	σ_h	1
Radar error	σ_I	0.1
Critical lower value for radii	$\mathbf{w_{crit}}$	0.4
Critical lower value for precipitation	$\mathbf{R_{crit}}$	0.1

The radar image at time 12:00 was used to initialize the model, and the last image in the sequence is at time 19:00, all together making up a sequence of eight images. In the real case, the mean of the filtered images and predictions are shown, not the last realization as for the synthetic case. The realizations were cut after 10000 iterations, using the last

5000 as the basis to do inference. This might be a bit exaggerated, but we want to be sure that the model has converged to yield good results. In order to reduce the run time, using a fewer number of realizations would be reasonable to consider.

5.2.1 Filtering

Figure 17 and 18 show the average of the filtered images to the right at time 13:00 to 18:00 over the geographical area shown in figure 16. Corresponding radar images are shown to the left. The image at time 13:00 is accurate in shape but is smoother than the observed radar images. There is also a small rain shower to the north west that filtered image does not capture. The image at 14:00, has a more detailed structure in the largest precipitation are, and we see that the model is able to build up the new precipitation showers entering the north west corner. This new filtered precipitation however, do not have a very detailed structure. We see that the filtered image over estimates the precipitation levels outside the boundaries at the moving precipitation field, as a result of the filtered image being very smooth. The advection looks reasonable, as the precipitation seems to move from north west,arriving from the North See. At the image at 15:00 there is also small levels of precipitation, but now at the north east corner, that is not captured by the filtered image. The trend is the same as for the previous two filtered images, that the precipitation fields that have been in the system for some time seem to get more detailed at each update.

At the top of figure 18, we see the radar and filtered image at time 16:00. The area of precipitation at north west is now the more detailed one, and the "old" one is seemingly dying out. The number of active basis functions seem to increase as time goes by, indicating that the model should be run a few times before we should inference about it. The advection seems reasonable as it points from north west, the direction where precipitation seems to enter. At time 17:00 the radar image shows precipitation moving "undiserable" from time 16:00, as the two radar images are not very similar. The one at time 17:00 is more smeared out. Still the filtered image looks satisfying, recreating and removing noise from the radar image. There is however parts with low intensities that do not show in the filtered image, but here the precipitation level is very small, below 1 mm. pr. hour. At time 18:00 the filtered image looks very much like the radar image, and the advection is pointing in a reasonable direction. It is also worth noticing that the small areas with the highest intensities are almost identical in the two images. The model can be said to focus on the most important rain showers, those which contains most precipitation, and not focus on areas with little and close to no precipitation.

5.2.2 Forecasting

Figure 19 show one hour forecast for time 14:00, 15:00 to 16:00 compared with the corresponding observed radar images. The forecast at 14:00 is of same shape as the filtered image at time 13:00, but can not predict the precipitation coming in from north west. Otherwise the forecast can be said to be good, showing a moving precipitation field that

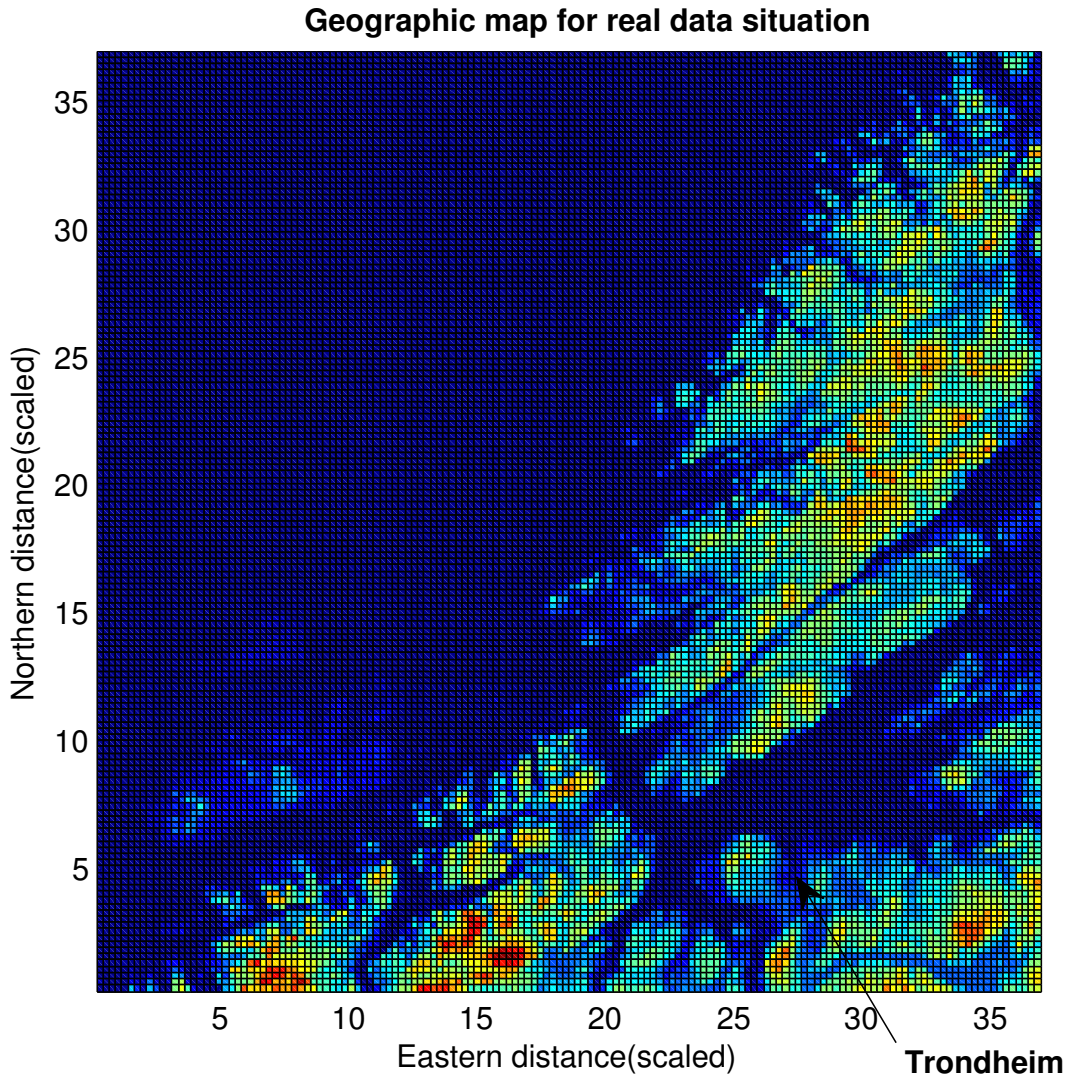


Figure 16: The geographical map of the real data situation. Trondheim is marked with an arrow. The axis are scaled to fit with the radar, filter and forecast images. The pixels are distanced 1000 m from each other, but each unit on the axis is of distance 4000 m.

is pretty accurate compared to the radar image. At time 14:00, there has been some decay of precipitation in the north east corner of the radar image, which our forecast could not predict. Also for time 15:00 we see that the forecast have difficulties with predicting the movement of the precipitation moving in from north west. But for all three predicted images in figure 19 we can see that the movement of the precipitation that has stayed in the grid for a couple of time steps are well tracked.

Figure 20 shows the one step forecasts for time 17:00, 18:00 and 19:00. At time 17:00 we see that there is an increasing number of active basis functions located at the north west corner. This is because the corresponding precipitation now have stayed in the grid for a couple of time steps, giving the model enough information to track the precipitation. To the next time step at 18:00, we see that the real advection has shifted a some, going from southern to a more eastern direction. The forecast then misses somewhat when estimating the new shape, but still we see that the outer contour of the field is generally satisfying. As time goes, and we get more basis functions, we see that the forecast gets less smooth. This is most likely a result of us holding the advection constant to the last realization in the MCMC algorithm presented in section 4.4.2, underestimating the variance and rely to much on this last realization when taking the model forward in time. However, the shape of the forecast precipitation field looks good, but it would be preferable that the image were smoother.

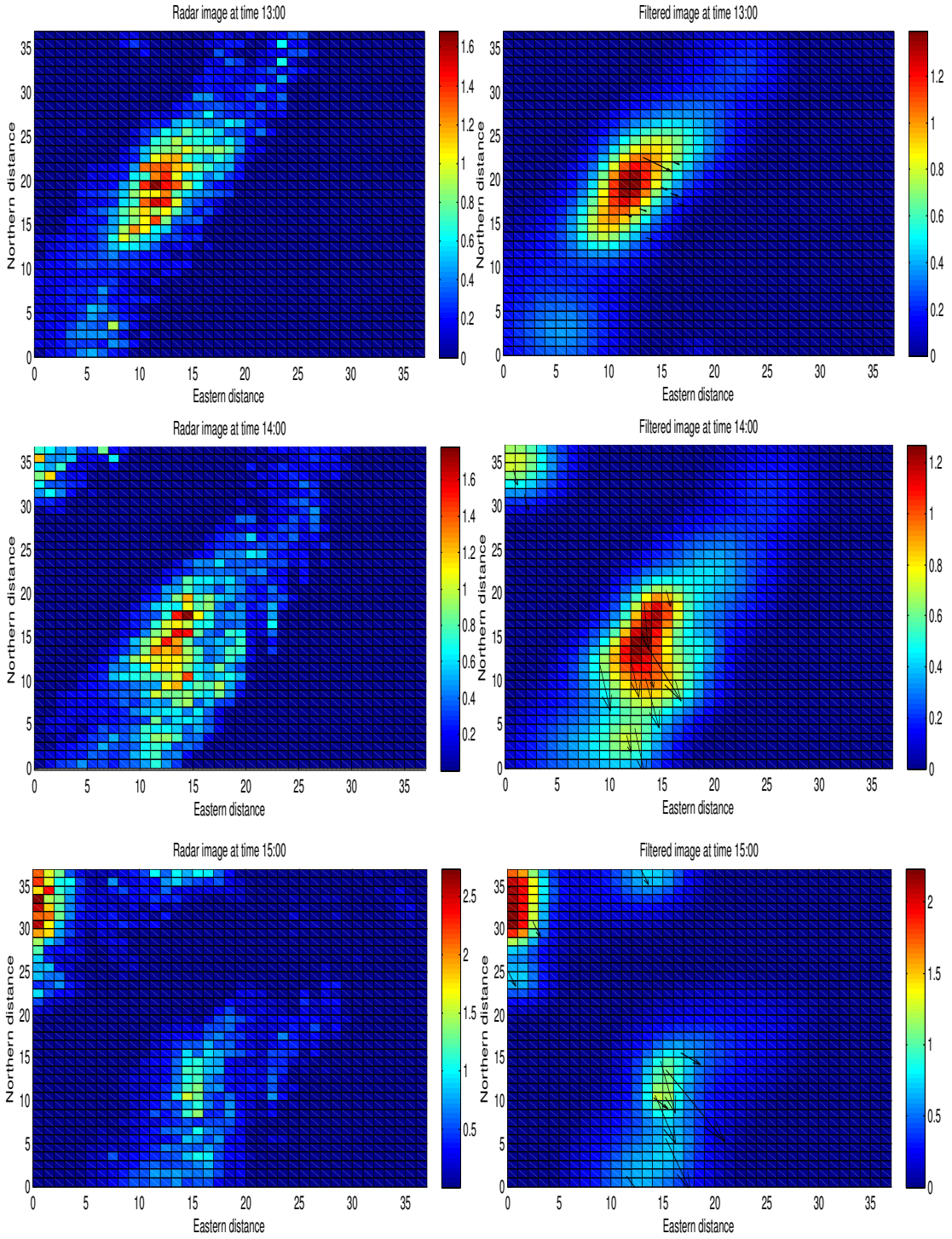


Figure 17: The filtered images(right) at time 13:00, 14:00 and 15:00 with corresponding radar images(left). Advection at active centers from last realization is shown with arrows. The filtered image at time 13:00 is the first update after the initialization, and is smoother than the radar image, indicating there is used too few basis functions. The filtered image at time 15:00 shows the build up of a new shower to the north west corner, but this shower is smother than the precipitation that have stayed in the grid from time 14:00. The filtered image at time 15:00 is the most detailed one, indicating that the model recreates the structure of the precipitation that have been in the grid for some time, better than new rain showers.

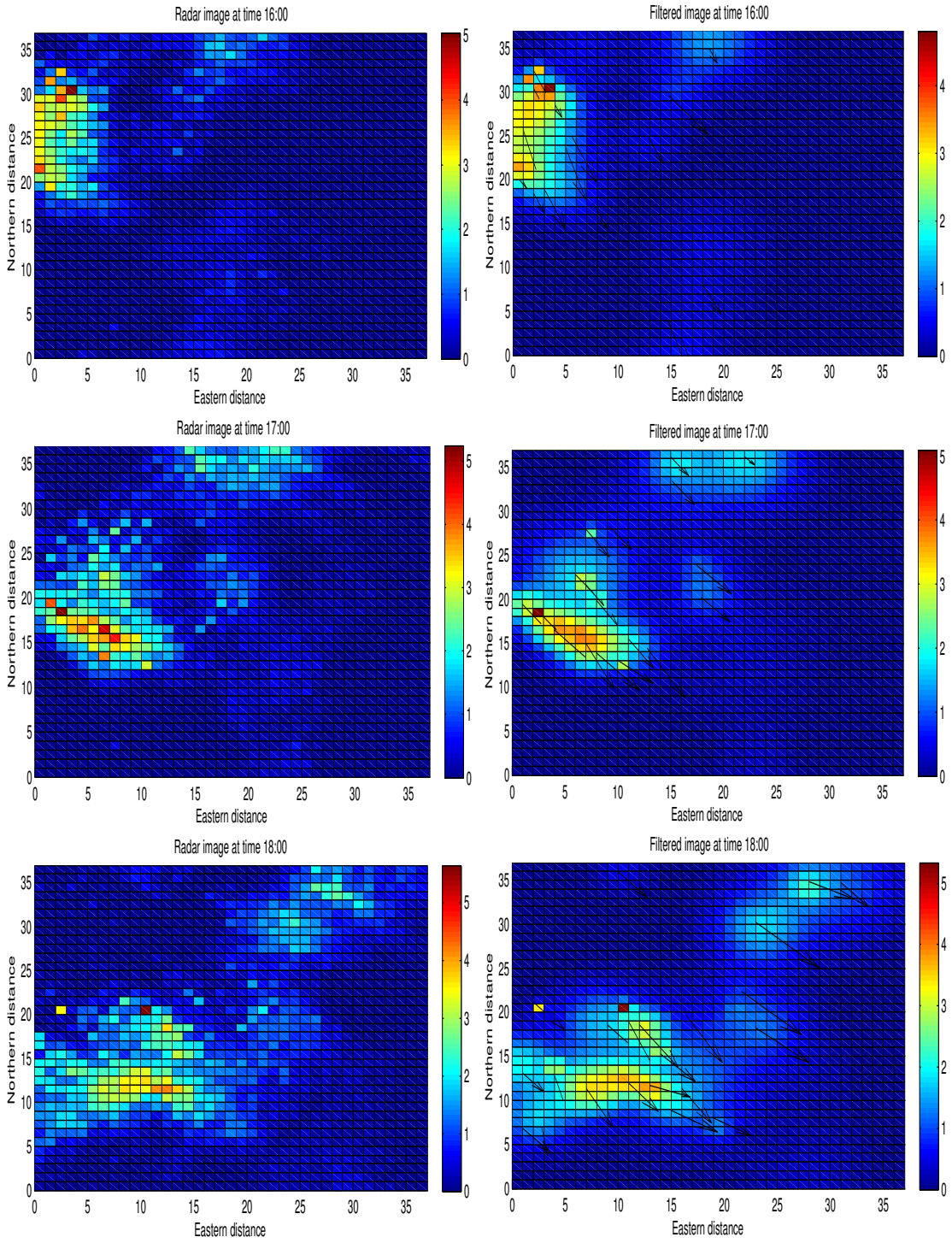


Figure 18: The filtered images(right) at time 16:00, 1700 and 18:00 with corresponding radar images(left). Advection at active centers from last realization is shown with arrows. As for figure 17 we see that the figures get more detailed, having more active basis functions as time passes. The filtered image at time 16:00 recreates the radar image well, having advection pointing to south east, the direction the precipitation seems to move. The filtered image at time 17:00 seems very reasonable, but misses some areas where the precipitation level is very light. The filtered image at time 18:00 is the most detailed one, as the model has been run for a few time steps. All intensities for the three filtered images are of same magnitude as the radar images.

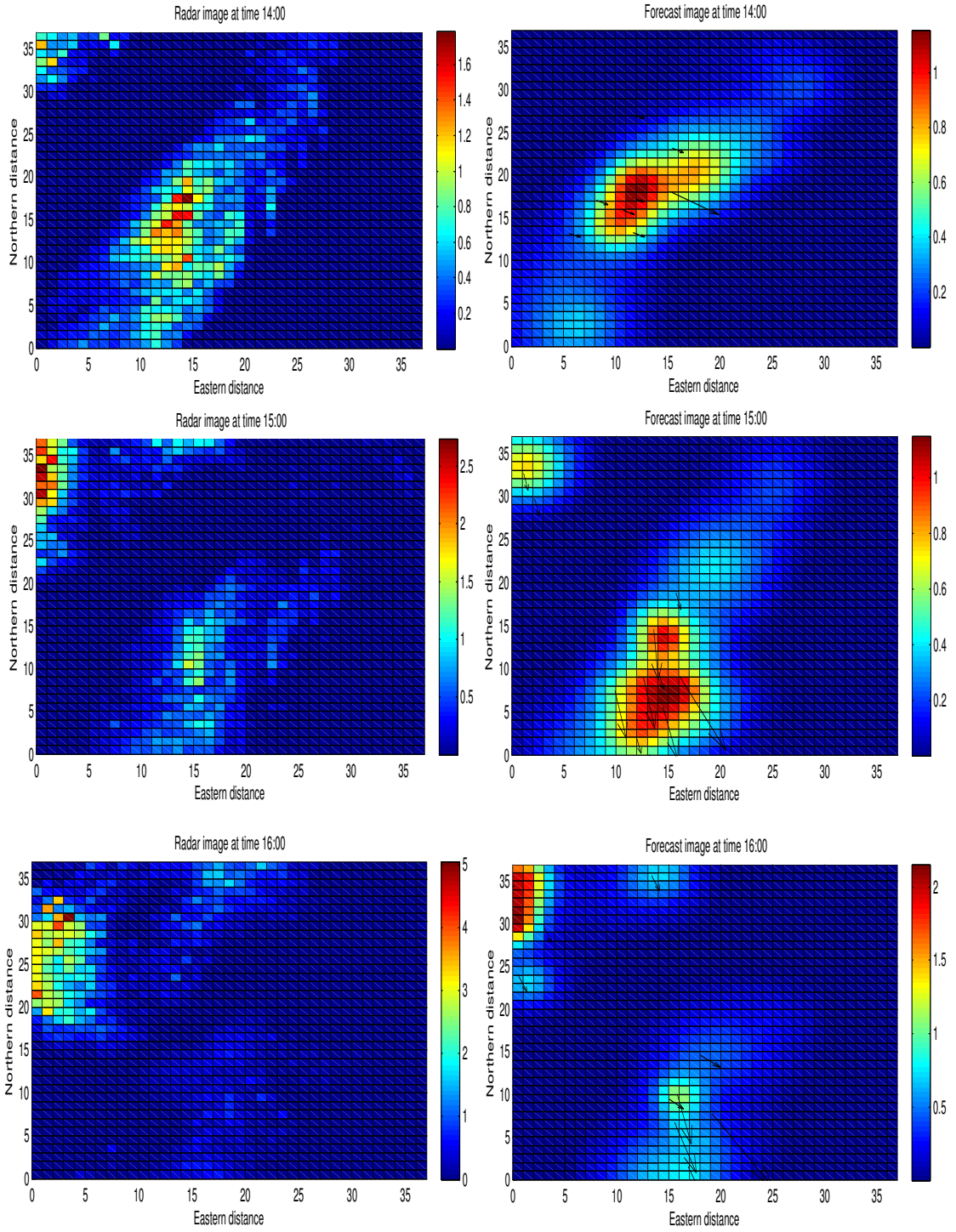


Figure 19: The average of the predicted images(right) at time 14:00, 1500 and 16:00 with corresponding radar images(left). The predictions are the mean of 5000 MCMC realizations from the one hour forecasting distribution. The advection illustrated is the last realization. The forecast at time 14:00 is smooth, but is similar in shape of the observed radar image. However, an underestimation of the advection, makes the prediction miss slightly at the location of the observed precipitation. The forecast at time 15:00 is able to predict precipitation at the north west corner, but fail to predict the decay of the precipitation located in the middle of the image. The forecast at time 16:00 looks at lot like the radar image at time 15:00, but fail to predict the decay of the precipitation in the middle and the increase at the north west corner.

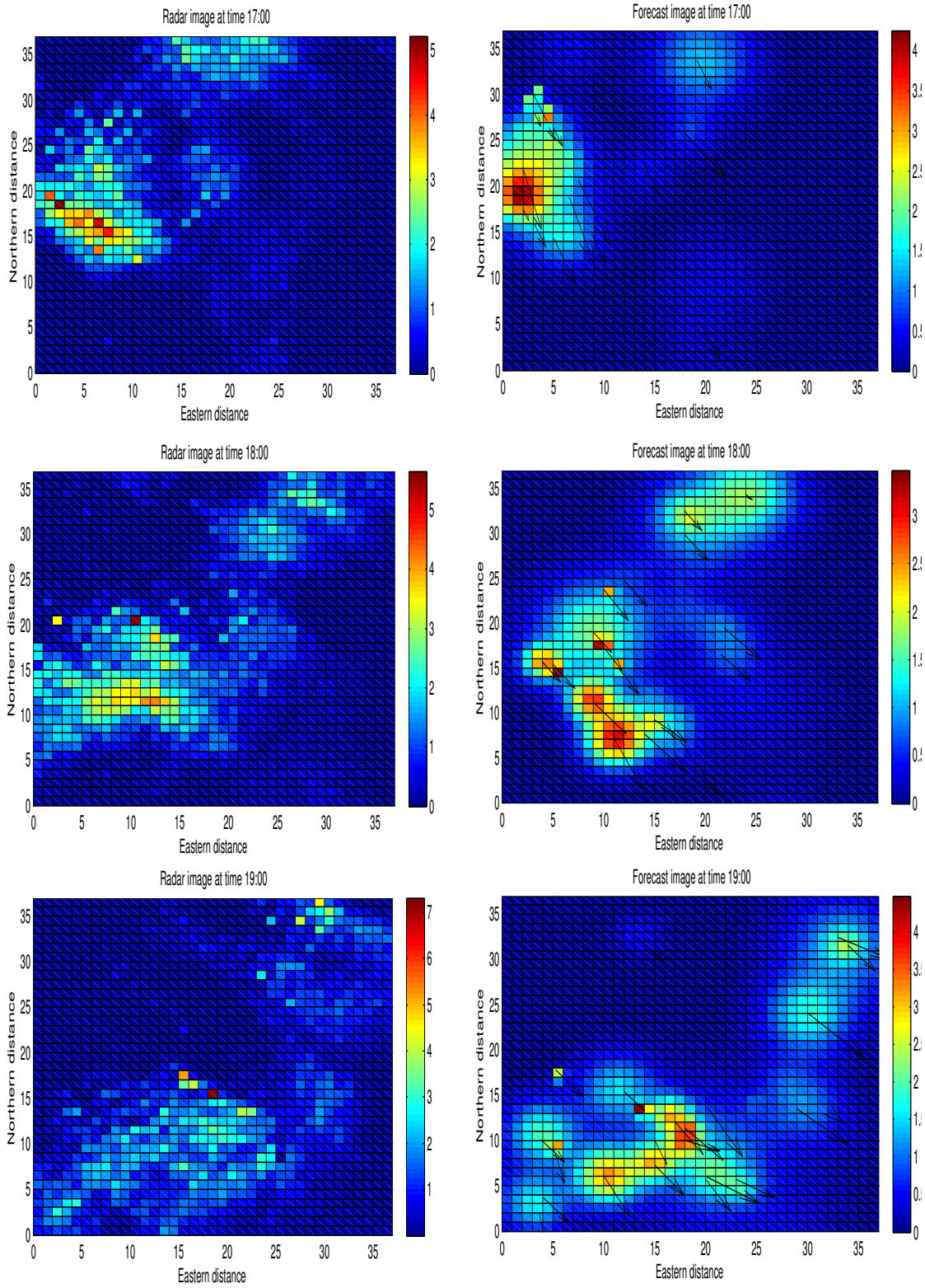


Figure 20: The predicted images(right) at time 17:00, 18:00 and 19:00 with corresponding radar images(left). The predicted image at time 17:00 is relatively good, but it is hard to predict the rapid changing shape and direction of the precipitation field. This is also the case for the forecast for time 18:00, that the model struggles to predict the rapid changing weather system. The forecast at time 19:00 is perhaps the best, but it would be preferable if all three forecasts were more smooth. This is the result of us only using the last realization when predicting, resulting in an underestimation of the variance.

5.2.3 Two hour precipitation forecasting for Trondheim

Looking at the radar image at time 17:00 from figure 20, we see that there is a precipitation field moving towards the city of Trondheim. The radar value at this time is 0, but we suspect that there might be some light rain showers moving towards the city. Assume we want to plan a head some weather sensitive outdoor activity in the hour from 18:00 to 19:00, and therefore want to predict when, and at which intensity, this precipitation will strike. A MCMC run with corresponding filtered values is assumed to be available at time 17:00.

We performs our predictions using estimators from section 4.4.2, and the current estimation for time 17:00 using the available filtered realizations. The results are available in table 3, with corresponding 99% confidence bands and radar observation. We use the strict criteria of 99% confidence bands due to the fact that we suspect to be underestimating the variance.The expected precipitation is calculated at every twentieth minute, from 18:00 to 19:00.

Table 3: Predicted precipitation values over Trondheim from 17:00 to 19:00

Time	lower 1 % percentile	Estimated value	upper 99 % percentile	Radar value
17:00	0	0.0050	0.1015	0
17:20	0	0.0080	0.1204	-
17:40	0	0.0134	0.1369	-
18:00	0	0.0332	0.1780	0.2221
18:20	0	0.0797	0.2213	-
18:40	0.	0.1397	0.2870	-
19:00	0.1169	0.1963	0.3879	0.4391

Looking at time 17:00, we see that the starting value, the last available filtered value, is close to the radar value. In fact we see that the first filtered value is below the critical lower value $\mathbf{w_{crit}} = 0.1$, which states that most realizations are zero. As time goes by, we notice that the expected amount of precipitation increase slightly the first hour, and increases up to almost 0.2 mm. pr. hour at time 19:00. The uncertainty also increases as a function of time, which is reasonable since longer forecasts are harder to predict. This was as expected, however we also notice that both radar values for time 18:00 and 19:00 are just outside the 99% confidence bounds. As discussed earlier in section 4.4.2, we know that we are probably underestimating the variance due to the fact that only the last realization of the advection is taken to the next level. In addition to this, something unexpected might have occurred which the model can not intercept. However, the expected values miss by about 0.19 mm. pr. hour at time 18:00, and by 0.24 mm. pr. hour at time 19:00, which could be said to be pretty accurate, giving a decent prediction as to how much rain is expected. The relative error might be said to be pretty high,

but we are capable to predict that there are incoming rain, that the intensity is not very high and that there is some uncertainty as to how much rain we could expect. It should also be stated that the observed radar values, which itself might not be correct, are close to the upper confidence bounds.

6 Conclusion

This paper defines a probabilistic model for short term forecasting using online radar images. The precipitation field is given as a sum of Gaussian basis functions, each consisting of a center, radius and an intensity. Using a sufficient number of such basis functions, any precipitation field can be created. The movement of each basis function is only governed by its corresponding advection, which is used to yield the forecast as a function of time. Labeling those basis functions that do not contribute to the final image as inactive, they can in turn be used to generate birth or death of rain showers. By assigning the basis functions and the advection prior distributions, the variables can be conditioned on a given radar image which is assumed to have independent normal distributed measurement errors. This update is done online via a approximated MCMC scheme which can account for birth and death of new rain showers.

Testing the model on data that are generated from the model itself, yield good results for both filtering and forecasting. Testing the model on real data also yield good filtered images, but the results from the prediction is mixed. The forecast images approximately predicts the weather, but it is hard to obtain very accurate predictions for a rapidly changing precipitation field. The confidence bounds obtained are assumed to be too narrow as we underestimate the uncertainty of the precipitation level due to the fact that the MCMC scheme is an approximated solution to the scheme presented in [7]. There are some weaknesses with the model, the most significant of these can be summarized as follows:

- We use an MCMC algorithm where only the last realization is taken to the next step, and the advection is held constant when estimating the forecast distribution. Ideally a routine that samples from the joint distribution of the previous and current state, of both the basis functions and the advection should be implemented. Without the joint sampling, and that only one value of the advection is used, the method does not account for the full uncertainty when forecasting.

- The time series used is measured at a rate of one hour. This is not ideal since the degree of change within the system might be too chaotic in this time frame. An arrival rate at 30 or 15 minutes would most likely improve the forecasting properties of the model.

- The hyper parameters of the system,the variances and correlation structure of the advection field, is specified using expert judgment. To obtain the radar error, [1] suggest to analyze the radar uncertainties on comparison against rain-gauge measurements. Using block-Kriging, [2] gives an online method to merge gauge data with radar images. The covariance matrices for the basis functions and advection can be estimated by adding an extra level to the hierarchical model, running an extensive MCMC run for many time steps offline. A routine that can account for a time dependent radar measurement error is presented in [9], combining radar

reflectivity with gauge data to deal with rapidly changing weather systems, each having a different drop size distribution.

- Using elliptical basis functions rather than circles, would be more flexible, which in turn would allow the precipitation field to be well represented with a smaller number of basis functions. This in turn might reduce the run time when updating the model at each time step.
- This model assumes that the advection has no dynamics, but there are many situations where the advection change rapidly or have more complex structure. Using the large scale information of the atmosphere from numerical weather forecasts when updating the advection, would most likely improve this update. However, lack of information about the error structure on the numerical weather forecasts would require further assumptions to be made [4].

References

[1] U. German, M. Berenguer, D. Sempere Torres, and M. Zappa. Real - Ensemble radar precipitation estimation for hydrology in a mountainous region. *QUATERLY JOURNAL OF THE ROYAL METEOROLOGICAL SOCIETY*, February 2009.

[2] E. Todini. A Bayesian technique for conditioning radar precipitation estimates to rain-gauge measurements. *Hydrology and Earth System Sciences*, 5:187–199, 2001.

[3] `http://energilink.tu.no/leksikon/vindkraft.aspx`.

[4] D. Cornford. A Bayesian state space modelling approach to probabilistic quantitive precipitation forecasting. *Journal of Hydrology*, 2003.

[5] C. B. Storli, T. C. M. Lee, J. Hanning, and D. Nycha. Tracking Of Multiple Merging And Splitting Targets: A Statistical Prespective. *Statistica Sinica*, 19:1–52, 2009.

[6] M. Fuentes, B. Reich, and G. Lee. Spatial-temporal mesoscale modelling of rainfall intensity using gage and radar data.

[7] S. K. Pang, J. F. Li, and S. J. Godsill. Sequential inference for dynamically envolving groups objects. University of Cambridge.

[8] `http://en.wikipedia.org/wiki/Weather_radar`.

[9] G. W Lee and I. Zawadzki. Radar calibration by gage, disdrometer, and polarimetry: theoretical limit caused by the variability of drop size distribution and application to fast scanning operational radar data. *Journal of Hydrology*, 2006.

[10] `http://en.wikipedia.org/wiki/Universal_Transverse_Mercator_coordinate_system`.

[11] `http://en.wikipedia.org/wiki/Meteorology`.

[12] `http://en.wikipedia.org/wiki/Primitive_equations`.

[13] F. M. White. *Fluid Mechanics*. McGraw-Hill, 2003.

[14] E. M. Ronquist. Incompressible Fluid Flow: Governing Equations. NTNU, January 2007.

[15] E. Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge University Press, 2003.

[16] I. Myrseth and H. Omre. The Ensamble Kalman filter and related filters. Norwegian University of Science and Technology, 09 08.

[17] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEE TRANSACTIONS ON SIGNAL PROCESSING*, 50(2), February 2002.

[18] S. M. Ross. *Introduction to Probability Models.* Elsevier Inc., 9 edition, 2007.